NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MODELING HELICOPTER BLADE DYNAMICS USING A MODIFIED MYKLESTAD-PROHL TRANSFER MATRIX METHOD

by

Juan D. Cuesta

December 1994

Thesis Advisor: E. Roberts Wood

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 1994	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Modeling Helicopter Blade Dynamics Using a Modified Myklestad-Prohl Transfer Matrix Method	5. FUNDING NUMBERS
6. AUTHOR(S) Cuesta, Juan D.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Rotor Blade vibratory stresses are of utmost importance in helicopter design. A modified Myklestad-Prohl method for rotating beams has been coded to assist in preliminary helicopter rotor blade design. The rotor blade dynamics program is part of the Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) program which was developed to aid in the preliminary design and analysis of helicopter rotor performance, stability and control, and rotor dynamics. JANRAD is an inter-active, user friendly program written in MATLAB® version 4.0 programming language and has been used extensively in the Naval Postgraduate School's capstone helicopter design course (AA 4306). A sample case is run and results are discussed

14. SUBJECT TERMS Helicopter Design, Rotor Blade Dynamics, Helicopter Vibrations, Myklestad-Prohl Method	15. NUMBER OF PAGES 77
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

MODELING HELICOPTER BLADE DYNAMICS USING A MODIFIED MYKLESTAD-PROHL TRANSFER MATRIX METHOD

by

Juan D. Cuesta
Lieutenant, United States Naval Reserve
B.S., State University of New York at Binghamton, 1986

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1994

Author: ______________________
Juan D. Cuesta

Approved by: ______________________
E. Roberts Wood, Thesis Advisor

____
S. K. Hebbar, Second Reader

____
Daniel J. Collins, Chairman
Department of Aeronautics and Astronautics

ABSTRACT

Rotor blade vibratory stresses are of utmost importance in helicopter design. A modified Myklestad-Prohl method for rotating beams has been coded to assist in preliminary helicopter rotor blade design. The rotor blade dynamics program is part of the Joint Army/Navy Rotorcraft Analysis and Design (JANRAD) program which was developed to aid in the preliminary design and analysis of helicopter rotor performance, stability and control, and rotor dynamics. JANRAD is an inter-active, user friendly program written in MATLAB® version 4.0 programming language and has been used extensively in the Naval Postgraduate School's capstone helicopter design course (AA 4306). A sample case is run and results are discussed.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. JANRAD 2

II. THEORY 5

A. HOLZER'S METHOD FOR TORSIONAL SYSTEMS 5

B. MODIFIED MYKLESTAD-PROHL METHOD 8

III. M-FILE DETAILS 15

A. DYNAM.M 15

B. BLADE.M 16

C. OUTPUT.M 22

IV. RESULTS 23

A. USER INSTRUCTIONS 23

B. OUTPUT 25

V. CONCLUSIONS AND RECOMMENDATIONS 43

A. CONCLUSIONS 43

B. RECOMMENDATIONS 43

APPENDIX A. DYNAM.M 45

APPENDIX B. BLADE.M 53

APPENDIX C. OUTPUT.M 57

APPENDIX D. H2 SAMPLE HELICOPTER 61

LIST OF REFERENCES 63

INITIAL DISTRIBUTION LIST 65

ACKNOWLEDGMENTS

I wish to thank my thesis advisor, Professor E. R. Wood, for his patience and guidance in completing this work. I also would like to acknowledge Maj. Robert K. Nicholson, Jr., USA and Maj. Walter M. Wirth, Jr., USA for their vision and efforts in creating JANRAD. I would also like to express my appreciation to Professor S. K. Hebbar for his assistance, and helpful comments on this work. Thanks also go to Lt. Dale Feddersen, USN, for his helpful advice and extensive work in upgrading JANRAD.

Finally, I would like to give my deepest thanks to my wife, Christina, for her support and encouragement, especially during these past two years.

I. INTRODUCTION

A. BACKGROUND

Dynamics play a major role in the design and development of the modern-day helicopter. Helicopter vibratory characteristics have been the decisive factor in the selection of winning designs in helicopter program competitions, as well as a major cause of cancellation of other programs. Low vibratory levels increase fatigue life of components, improve passenger comfort, and decrease crew fatigue. Therefore, it is of utmost importance to be able to analyze and examine the effect of these vibrations in the preliminary stages of helicopter design.

In helicopters, vibrations fall into three categories: vibrations due to rotor excitation, which occur at frequencies that are integral multiples of the rotor's rotational speed; vibrations due to random aerodynamic excitation, where the frequencies observed are the natural frequencies of the structure being excited, and self-excited vibrations such as flutter and ground resonance. In this paper the primary source of helicopter vibrations will be considered, that is, the rotor system.

The alternating forces that excite the rotor blade are almost entirely due to the periodic variations in blade airloads encountered in flight. These time-dependent aerodynamic forces occur due to the variations in velocity and angle of attack encountered by the rotating blade. In low-speed flight blade-vortex interaction plays a

significant role in blade angle of attack variation due to the interaction of a rotor blade with the trailing tip vortex shed by the preceding blade. In steady-state level flight these loads can be considered harmonic since they occur at multiples of the rotor's rotational speed, i.e. 1/rev., 2/rev., 3/rev., etc. The blade's ensuing response to these loads is transferred as vibratory shears and moments to the rotor head and filter through to the fuselage. Therefore, the ability to analyze the stresses in the helicopter's rotor blade will lead to an understanding of how to alleviate these stresses. To that end, this thesis presents a computational tool to examine the flatwise dynamic loads of a rotor blade with varying material properties.

B. JANRAD

The **J**oint **A**rmy/**N**avy **R**otorcraft **A**nalysis and **D**esign (JANRAD) computer program was developed to meet the specific needs of the Naval Postgraduate School (NPS) for preliminary helicopter design. JANRAD was written as an interactive, user friendly program, capable of accurately and quickly solving helicopter design problems at the preliminary design level. JANRAD consists of three major subroutines. The first routine, JANRAD Performance, calculates the trim solution and various performance parameters used in helicopter design. This program is described in detail in Ref. 1. The second component is JANRAD Stability. This routine calculates stability derivatives, and describes the open loop control characteristics using a linear state space model. The details of this program are provided in Ref. 2.

The third component is JANRAD Dynamics, written by the author, which adopts the same user friendly menu driven format to assist in determining preliminary rotor blade design parameters. The program uses a modified Myklestad-Prohl Method [Ref. 3] to determine the uncoupled, flatwise, forced response of the helicopter rotor blade to aerodynamic loads and dynamic forces. These forces are calculated using the same helicopter design parameters and data output from the JANRAD Performance routine, making JANRAD a truly integrated design tool.

JANRAD was originally written for version 3.5 of 386-MATLAB® but since has been updated to MATLAB® version 4.0 for WINDOWS®. MATLAB® is a high performance interactive software package for scientific and engineering numeric computations. This program was chosen due to its wide accessibility at the NPS and for its ease of use. This frees the engineering student from the burden of learning "low level" programming languages and allows the user to concentrate on the design process itself. Although there are other helicopter design programs such as HESCOMP, CAMRAD or RACAP; these programs are often proprietary, expensive to acquire, and cumbersome to use. Whereas the general simplicity of MATLAB® allows the NPS student to modify JANRAD to meet his/her specific needs.

II. THEORY

There are several methods for determining the forced response of dynamic systems. These methods may be classified into two categories: exact methods and approximate methods. In exact methods an explicit solution of the governing differential equation is obtained in closed form. In general, explicit solutions are possible only for linear differential equations with constant coefficients. These types of equations are associated with small amplitude vibrations of structures having uniformly distributed mass and stiffness properties. When these properties are not uniformly distributed, the coefficients are variable and exact solutions are not always possible. These problems in transverse vibrations of beams can be solved by an approximate numerical procedure which was developed independently by Prohl [Ref. 4] and Myklestad [Ref. 5]. As a historical, as well as a pedagogical precursor to the Myklestad-Prohl method, the Holzer method [Ref. 6, Ref. 7] demonstrates a simpler lumped mass system that models torsional dynamics.

A. HOLZER'S METHOD FOR TORSIONAL SYSTEMS

The Holzer method is an approximate solution to the second order differential equation of a uniform slender shaft:

$$\frac{d^2\theta}{dx^2} + \frac{\omega^2 I}{GJ}\theta = 0 \qquad (1)$$

where

θ = maximum amplitude of torsional vibration

ω= frequency of a natural mode

I = mass moment of inertia per unit length

GJ = torsional stiffness

The Holzer method approximates the differential equation of motion by two difference equations which replace the distributed structure with one in which the masses and connecting elastic elements are discretized or "lumped." These numerical models are often referred to as *discrete* or *lumped mass systems*. Figure 1 shows a part of the non-uniform shaft, lumped mass system used in deriving the Holzer difference equations.

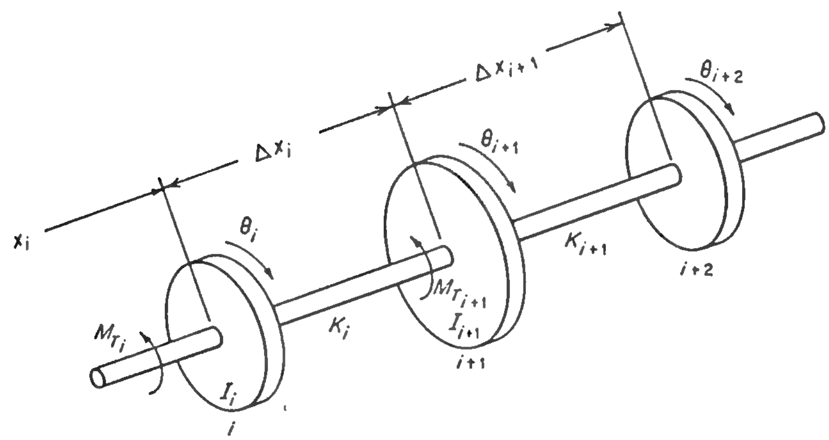

Figure 1. Ideal Lumped Mass System For A Non-Uniform Shaft From Ref. 7

The equation of moment equilibrium for the *i*th mass is:

$$M_{T\,i+1} - M_{T_i} + \omega^2 I_i \theta_i = 0 \tag{2}$$

where

$M_{T_{i+1}}$ = torque applied to the mass $i + 1$ by shaft section i

M_{T_i} = torque applied to the mass i by shaft section i - 1

I_i = moment of inertia of mass i

θ_i = angular displacement of mass i

ω = angular frequency of vibration

A second equation is derived by expressing the twist of the shaft section i in terms of the torque on that section.

$$\theta_{i+1} - \theta_i = \frac{M_{T\,i+1}}{k_i} \tag{3}$$

where

k_i = stiffness of shaft section i

Equations 2 and 3 can be written in the following form for purposes of an iterative analysis

$$M_{T\,i+1} = M_T - \omega^2 I_i \theta_i \tag{4}$$

$$\theta_{i+1} = \theta_i + \frac{M_{T\,i+1}}{k_i} \tag{5}$$

Equations 4 and 5 may also be derived from the differential equation for a uniform slender shaft, Equation 1. The parametric form of Equation 1 may be written as

$$\frac{dM_T}{dx} = -\omega^2 I(x)\,\theta \tag{6}$$

and

$$\frac{d\theta}{dx} = \frac{M_T}{GJ(x)} \tag{7}$$

If the first derivatives are approximated by the difference forms as

$$\frac{dM_T}{dx} \approx \frac{M_{T\,i+1} - M_{T\,i}}{\Delta x_i} \tag{8}$$

$$\frac{d\theta}{dx} \approx \frac{\theta_{i+1} - \theta_i}{\Delta x_i} \tag{9}$$

and by using the definitions

$$I_i = I(x)\,\Delta x_i \tag{10}$$

$$k_i = \frac{GJ(x)}{\Delta x_i} \tag{11}$$

we may once again obtain the Holzer difference Equations 4 and 5.

B. MODIFIED MYKLESTAD-PROHL METHOD

An extension of the Holzer method for torsional systems is the modified Myklestad-Prohl method for lumped mass beam systems [Ref. 8]. While the Holzer method solves a second order linear equation with two difference equations, the Myklestad-Prohl method solves the fourth order differential equation for a uniform beam:

$$EI\frac{d^4Z}{dx^4} = -m\,\frac{d^2Z}{dt^2} \tag{12}$$

with four difference equations, where

Z = deflection, expressed in terms of vertical displacement

dx = spanwise differential length

m = beam mass per unit length

EI = beam stiffness or bending modulus

The modified Myklestad method uses these four difference equations to progressively compute the shear, moment, slope, and deflection from one section to the next, taking the centrifugal and aerodynamic forces into account. Figure 2 illustrates an element of the blade lumped mass model with the uncoupled flatwise dynamic forces and moments acting on it.

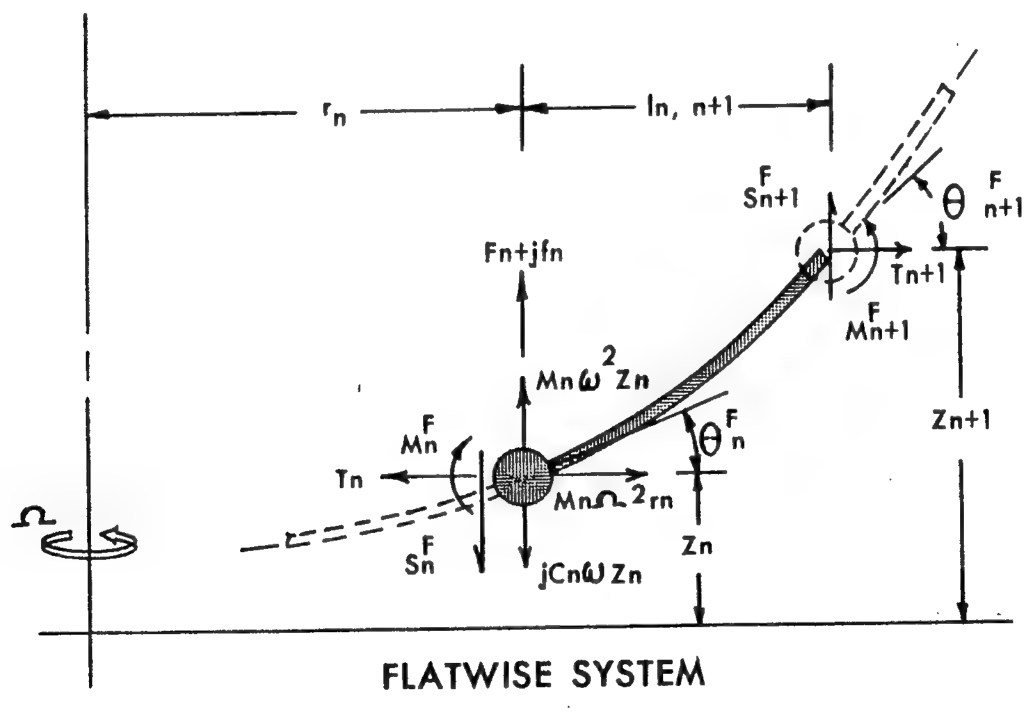

Figure 2. Blade Element Lumped Mass Model For Uncoupled Flatwise Vibration From Ref. 3

By taking the blade element model into consideration it is possible to write equations for the tension, shear, and moment at n entirely in terms of quantities at $n+1$. These can then be substituted into the geometric equations for the slope, θ and the vertical deflection, Z. Writing the equations of equilibrium for the *n*th element of the uncoupled flatwise system we have:

Centrifugal Force:

$$T_n = T_{n+1} + m_n \Omega^2 r_n \tag{13}$$

where:

T_n = centrifugal tension force acting on the *n*th blade element

m_n = mass of *n*th blade element

Ω = rotor angular velocity

r_n = distance of the *n*th blade element from the axis of rotation

Shear:

$$S_n = S_{n+1} + m_n \omega^2 Z_n - jC_n \omega Z_n + F_n + jf_n \tag{14}$$

where:

S_n = flatwise shear force acting on the *n*th blade element

m_n = mass of *n*th blade element

ω = applied excitation frequency

Z_n = deflection, expressed in terms of vertical displacement of the *n*th blade element

$F_n + jf_n$ = aerodynamic thrust acting on the *n*th blade element for a particular frequency component

C_n = flatwise aerodynamic damping for the *n*th blade element

The aerodynamic damping term can be expressed by:

$$C_n = \frac{dC_L}{d\alpha}(chord)_n(l_{n,n+1})(\rho/2)(\Omega r_n) \tag{15}$$

where:

$\frac{dC_L}{d\alpha}$ = lift coefficient slope for two-dimensional blade section

$l_{n,n+1}$ = length of *n*th blade element

ρ = mass density of air

Ω = rotor angular velocity

r_n = distance of the *n*th blade element from the axis of rotation

Additionally, from Figure 2, we have the equations for the moment:

$$M_n = M_{n+1} + S_{n+1}\, l_{n,n+1} - T_{n+1}(Z_{n+1} - Z_n) \tag{16}$$

where:

M_n = flatwise bending moment at *n*th blade station

S_n = flatwise shear force acting on the *n*th blade element

$l_{n,n+1}$ = length of *n*th blade element

T_{n+1} = centrifugal tension force acting on the *n+1* blade element

Z_n = deflection, expressed in terms of vertical displacement of the *n*th blade element

slope:

$$\theta_n = \theta_{n+1}\left(1 + T_{n+1}\frac{l^2_{n,n+1}}{2EI}\right) - M_{n+1}\frac{l_{n,n+1}}{EI} - S_{n+1}\frac{l^2_{n,n+1}}{2EI} \tag{17}$$

where:

θ_n = flatwise slope of the elastic axis at the nth blade station

T_n = centrifugal tension force acting on the nth blade element

$l_{n,n+1}$ = length of nth blade element

EI = beam bending modulus at the nth blade station

M_n = flatwise bending moment at nth blade station

S_{n+1} = flatwise shear force acting on the $n+1$ blade element

and deflection:

$$Z_n = Z_{n+1} - \theta_n l_{n,n+1} + \frac{T_{n+1}\,\theta_{n+1}\,l^3_{n,n+1}}{3EI} - \frac{M_{n+1}\,l^2_{n,n+1}}{2EI} - \frac{S_{n+1}\,l^3_{n,n+1}}{3EI} \tag{18}$$

where:

Z_n = deflection, expressed in terms of vertical displacement of the nth blade element

θ = flatwise slope of the elastic axis at the nth blade station

$l_{n,n+1}$ = length of nth blade element

T_n = centrifugal tension force acting on the nth blade element

EI = beam bending modulus at the nth blade station

M_{n+1} = flatwise bending moment at $n+1$ blade station

S_{n+1} = flatwise shear force acting on the $n+1$ blade element

The equations, Equations 14,16,17, and 18, may be rewritten so that all the terms on the right side of the equation are in terms of $n+1$ variables. By then expressing

these equations in matrix form and integrating the blade elements from tip to root , the shear, moment, slope, and deflection at the blade root may be expressed in terms of corresponding unknown tip values. With the use of the appropriate root boundary conditions the unknown tip values may be determined. Once this is done the shear, moment, slope, and deflection can be determined along the length of the rotor blade. The details of this procedure will be discussed in the *Blade.m* program details section.

III. M-FILE DETAILS

A. DYNAM.M

The first module of the blade dynamic analysis portion of JANRAD is *Dynam.m*. The primary function of this subroutine is to enter and store rotor blade data required for analysis. Since this portion of JANRAD uses variables which are cumbersome to keep re-entering, it also follows a similar menu driven format that allows the user to save variables for later analysis. *Dynam.m* uses output data from the JANRAD Rotor Performance program so it must first determine whether it has been run. If it determines that the Rotor Performance has not been run, the user is exited out of the program to enter the JANRAD main menu again. The program then asks the user to either edit or create a data file. If editing is chosen, the program then asks the user for the file name. If the file is not found, the user is instructed to try again or exit the program. When the file is found, an edit menu is presented where the user may change the rotor blade parameters.

Whether the user chooses to create a new data file or to edit a previous one, the program prompts the user to enter the rotor blade material properties and weight distribution. The most frequently applied boundary conditions for the root end of the rotor blade are either articulated or hingeless. For the articulated case, the flatwise moment is zero, whereas in the hingeless case the root slope is equal to zero. In both cases the blade root vertical displacement is zero.

The material properties may be entered in two ways, either as a combination of the modulus of elasticity, E (lb/in^2), and the rotor blade flapwise moment of inertia, I_b (in^4), or in combination as the rotor blade stiffness coefficient, EI ($lb \cdot in^2$). These variables along with the weight distribution are entered as vectors whose elements represent the elemental properties along the blade. The user is instructed to enter the rotor blade properties from blade tip to blade root and is reminded to enter the same number of blade elements as was previously chosen in the main JANRAD menu. When entering the weight distribution, the weight is given as previously entered for the Rotor Performance portion of JANRAD. The entered data vectors are then checked for correct length and stored. The data is saved using a file name in the same manner as used in the main JANRAD program. These specifics are outlined in Ref. 1.

B. BLADE.M

Blade.m uses the Mylestad-Prohl equations for rotating beams in complex form to calculate the flatwise forced response of a rotor blade of varying material properties. This process is accomplished in four steps:

1. Calculate the aerodynamic damping, the centrifugal force, and the steady and first ten harmonic thrust differentials.
2. Calculate the transfer matrices for the harmonic blade response along the length of the blade.
3. Solve for the rotor blade tip unknowns (Z_T, θ_T).
4. Repeat steps 3 and 4 utilizing the respective thrust harmonic and excitation frequency.

Before discussing the four step process used in the program, we must return to the Myklestad-Prohl equations and present them in the form they are used in *Blade.m*.

The modified Myklestad equations, which were introduced earlier as Equations 14, 16 - 18, may be rewritten so that all the $n-1$ terms are on the right side of the equations, as follows:

$$S_n = d_{11} S_{n+1} + d_{12} M_{n+1} + d_{13} \theta_{n+1} + d_{14} Z_{n+1} \tag{19}$$

$$M_n = d_{21} S_{n+1} + d_{22} M_{n+1} + d_{23} \theta_{n+1} + d_{24} Z_{n+1} \tag{20}$$

$$\theta_n = d_{31} S_{n+1} + d_{32} M_{n+1} + d_{33} \theta_{n+1} + d_{34} Z_{n+1} \tag{21}$$

$$Z_n = d_{41} S_{n+1} + d_{42} M_{n+1} + d_{43} \theta_{n+1} + d_{44} Z_{n+1} \tag{22}$$

where d_{ij} between station n and station $n+1$ is given by:

$$d_{11} = 1 + (m_n \omega^2 - jC_n \omega) \frac{l^3_{n,n+1}}{6EI} \qquad d_{12} = \frac{l^2_{n,n+1}}{2EI} (m_n \omega^2 - jC_n \omega) \tag{23}$$

$$d_{13} = -(m_n \omega^2 - jC_n \omega) \left(\frac{T_{n+1} l^3_{n,n+1}}{6EI} + l_{n,n+1} \right) \qquad d_{14} = (m_n \omega^2 - jC_n \omega) \tag{24}$$

$$d_{21} = \frac{T_{n+1} l^3_{n,n+1}}{6EI} + l_{n,n+1} = -d_{43} \qquad d_{22} = \frac{T_{n+1} l^2_{n,n+1}}{2EI} + 1 = d_{33} \tag{25}$$

$$d_{23} = -\left(\frac{T^2_{n+1} l^3_{n,n+1}}{6EI} + T_{n+1} l_{n,n+1} \right) \qquad d_{24} = 0 = d_{34} \tag{26}$$

$$d_{31} = -\frac{l^2_{n,n+1}}{2EI} = -d_{42} \qquad d_{32} = -\frac{l_{n,n+1}}{EI} \tag{27}$$

$$d_{41} = \frac{l^3_{n,n+1}}{6EI} \qquad d_{44} = 1 \tag{28}$$

These equations may also be written in the matrix form:

$$\{\zeta\}_n = [D_{n+1}]\{\zeta\}_{n+1} + \{\alpha_\zeta\}_{n+1} \tag{29}$$

where

$$\{\zeta\}_n = \begin{bmatrix} S \\ M \\ \theta \\ Z \end{bmatrix}_n \tag{30}$$

$$D_{n+1} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix}_{n+1} \tag{31}$$

and

$$\{\alpha_\zeta\}_{n+1} = \begin{bmatrix} \alpha_S \\ \alpha_M \\ \alpha_\theta \\ \alpha_Z \end{bmatrix}_{n+1} \tag{32}$$

Matrix Equation 29 relates the response of the $n+1$ element to the adjacent inboard nth element through the transfer matrix $[D]_{n+1}$. It should be noted that $[D]_{n+1}$ is dependent only on known or pre-calculated parameters. The elements in Equation 32 represent the

shear, moment, slope, and deflection which are solely due to aerodynamic loads. Now that the Myklestad-Prohl equations have been presented in the matrix form used in the program, we will proceed to discuss the first procedure of *Blade.m*.

The flatwise aerodynamic damping term, C_n, for the *n*th term is given by:

$$C_n = \frac{dC_L}{d\alpha}(chord)_n(l_{n,n+1})(\rho/2)(\Omega r_n) \tag{33}$$

where

$\frac{dC_L}{d\alpha}$ = lift coefficient slope for two-dimensional blade section

$l_{n,n+1}$ = length of *n*th blade element

ρ = mass density of air

Ω = rotor angular velocity

r_n = distance of the *n*th blade element from the axis of rotation

The lift coefficient slope, $\frac{dC_L}{d\alpha}$, as well as the mass density of air, ρ, and the rotor angular velocity, Ω, have been previously entered as required variables for JANRAD rotor performance. The elemental chord length has been updated to account for taper ratio and is also taken from JANRAD.

In calculating the steady and first ten harmonic thrust differentials the elemental thrust forces along the blade are obtained from the rotor performance analysis portion of JANRAD. Using a numerical Fourier analysis, these thrust forces are resolved into a steady and ten harmonic components. These forces may be expressed in complex form as:

$$F_\psi(r_n) = F_0(r_n) + \sum_{i=1}^{10} \Re eal(f_i(r_n) - jF_i(r_n))e^{jn\psi} \tag{34}$$

where the steady aerodynamic thrust is given by:

$$F_0(r_n) = \frac{\sum_{j=1}^{n_\psi} dT_\psi(r_n)}{n_\psi} \tag{35}$$

and the harmonic thrust components are:

$$F_i(r_n) = \frac{2}{n_\psi} \sum_{j=1}^{n_\psi} dT_\psi(r_n) \sin(i\psi) \qquad i = 1, 2 \ldots 10 \tag{36}$$

$$f_i(r_n) = \frac{2}{n_\psi} \sum_{j=1}^{n_\psi} dT_\psi(r_n) \cos(i\psi) \qquad i = 1, 2 \ldots 10 \tag{37}$$

where:

n_ψ = number of azimuth sectors

$dT_\psi(r_n)$ = elemental thrust at blade azimuth position ψ, and blade station r_n

ψ = azimuth angle position

i = ith harmonic component

The second step in *Blade.m* involves progressively integrating Equation 29 from tip to root. The general form of Equation 29 may be obtained prior to substituting the boundary conditions for either end of the blade. In this manner the shear, moment, slope, and deflection at the blade root may be expressed in terms of the corresponding unknown tip values. The form of Equation 29 that expresses the blade root values in terms of the corresponding unknown tip values is given as:

$$\{\zeta_0\} = [D_0]\{\zeta_T\} + \{\alpha_{\zeta 0}\} \tag{38}$$

In Equation 38, the variables are as defined as given in Equations 30 - 32, except that the subscript 0 denotes the root variable and the subscript T, the tip variable. The column matrices of shear, moment, slope, and deflection are complex, as is the force or $\{\alpha_{\zeta 0}\}$ column matrix. Constants in the 4 x 4 blade matrix, $[D_0]$, are also complex. These values result from the integration along the entire blade and reflect frequency as well as blade mass and stiffness properties.

In the next step we solve for the unknown blade boundary conditions. Since we are dealing only with uncoupled flatwise vibrations, two boundary conditions must be satisfied at each end of the blade. Application of the known tip boundary conditions eliminates two unknowns in Equation 38 resulting in a 2 x 2 $[D_0]$ blade matrix equation, which is then solved for the unknown tip values: (Z_T, θ_T).

For conventional free-ended helicopter blades, tip shears and tip moments are zero. The root boundary conditions are determined by the root construction of the blade, either articulated or rigid (hingeless). In either case the root vertical displacement is zero, but the articulated blade has zero moment whereas the rigid blade has zero slope at the root. Solution of the blade's forced harmonic response is then obtained by 1) applying appropriate root boundary conditions, 2) inverting the reduced 2 x 2 complex blade matrix $[D_0]$, 3) solving for the unknown flatwise tip slope and deflection, and 4) using the resulting 4 x 4 $[D]_n$ to solve for the shear, moment, slope, and deflection along the length of the blade. *Blade.m* uses the 2 x 2 matrix inversion formula instead of relying on the

MATLAB® function. This is to reduce the possibility of numerical errors. Finally, this procedure is repeated (for a total of eleven times) for the steady (zeroth) through tenth harmonic blade response with excitation applied successively at corresponding excitation frequencies ($0, \Omega, 2\Omega, 3\Omega, \ldots, 10\Omega$). The total response of the blade at each azimuth position is then obtained by superposition of these results.

C. OUTPUT.M

This M-file computes the rotor blade total response and displays the results along with the steady and first ten harmonic responses. *Output.m* also displays graphically the blade material properties. These display options are available from a menu driven format and the graphs may be printed from the File Menu of the figure Window®.

The rotor blade total response at each blade station is calculated from the steady and harmonic responses and is given by:

$$\begin{bmatrix} S \\ M \\ \theta \\ Z \end{bmatrix}_{Total} = \begin{bmatrix} S \\ M \\ \theta \\ Z \end{bmatrix}_{Steady} + \sum_{i=1}^{10} \left(\Re eal \left\{ \begin{bmatrix} S \\ M \\ \theta \\ Z \end{bmatrix}_i \right\} \sin(i\psi) + \Im mag \left\{ \begin{bmatrix} S \\ M \\ \theta \\ Z \end{bmatrix}_i \right\} \cos(i\psi) \right) \tag{39}$$

The total response may be viewed as a mesh plot for all azimuth angle positions or at individual angles. The weight and flatwise stiffness distribution as a function of blade radius, r, may also be displayed. Examples of sample output may be viewed in the Results and Recommendations section of this thesis.

IV. RESULTS

A. USER INSTRUCTIONS

The Rotor Dynamic Analysis program is a major subroutine of the main JANRAD program, and can be run after initiating the main program. JANRAD main program installation, execution, and input requirements may be found in Refs. 1 and 2. Figure 3 shows the main Execution Menu from where the Rotor Dynamic Analysis program may be accessed. This menu is accessed after either entering a new data file or editing a

```
      *** EXECUTION MENU ***

  1. Rotor Performance Analysis
  2. Stability and Control Analysis
  3. Rotor Dynamic Analysis
   (Rotor Performance Analysis should be run first)
  4. Change data
  5. Quit

 Enter a 1, 2, 3, 4, or 5   >>
```

Figure 3. JANRAD Main Execution Menu

previously stored data file. It should be noted that since the Rotor Dynamic Analysis program uses output data from the Rotor Performance Analysis portion of JANRAD, the performance analysis should be run first. If the Rotor Performance Analysis has already been run for a previously entered data file (and all input parameters remain the same), and a ".mat" and "_p.mat" already exist for that file, the user may proceed directly to the Rotor Dynamic Analysis program.

The data entry and save procedures for the Rotor Dynamic Analysis follow similar formats to those of Refs. 1 and 2. The first screen of the Blade Dynamic subroutine prompts the user to either edit a previously entered data file or create a new file. If the edit file option is chosen, you will be prompted to enter the file name without the extension. Just as in the other JANRAD subroutines, if the file name is not found the Rotor Dynamic program will allow the user to try again. Once the file has been loaded, the Blade Dynamics edit menu, as shown in Figure 4, will be displayed.

```
***BLADE DYNAMICS EDIT MENU ***

1. root boundary condition
2. blade material properties
3. weight distribution

*** ALL OTHER BLADE INFORMATION IS ENTERED IN MAIN JANRAD MENU ***

0. NO CHANGES
Input the parameter to change:
```

Figure 4. Blade Dynamics Edit Menu

Whether the rotor blade input parameters are being edited or a new file created the data is entered in the same manner. In the file edit mode, if you choose not to change the value, pressing <Enter> will re-enter the previous value. In the file create mode, the user will continue to be prompted to enter a value, if none has been entered, or if a variable of incorrect dimensions is entered. For the root boundary condition, the screen will prompt the user to choose the root boundary condition that corresponds to the construction of the blade. The user may enter the boundary conditions for an articulated rotor blade. When

entering the rotor blade material properties the user has the option of either entering the rotor blade variable stiffness (EI, lbs·in^2), or variable modulus of elasticity (E, lbs/in^2) and variable moment of inertia (I_{xx}, in^4) separately. In either case the screen will prompt the user to enter the row vector variable starting from the tip of the blade and ending with the root; it will also give an example along with the correct units. It is noted that while I_{xx} typically varies with radius, in most cases, E is constant. The program also reminds the user of the number of elements the row vectors must have. The number of blade elements is a variable that was entered for the Rotor Performance Analysis portion of JANRAD. The weight distribution is entered in the same manner. In the file edit mode the user is returned to the edit menu after editing any selection until no further changes are required.

Once the data file has been edited or created the program will prompt the user for a file name. If the user is in the edit mode, and wishes to keep the same file name they can press <Enter> and the same file name will be used. Upon selecting a blade file name the Rotor Dynamic Analysis program will save the file with a ".mat" extension and calculate the flatwise forced response of the entered blade.

B. OUTPUT

The sample output is from a helicopter that is based on the Sikorsky CH-53A which has a six bladed, articulated rotor system. It is taken from "*Parametric Investigation of the Aerodynamic and Aeroelastic Characteristics of Articulated and Rigid (Hingeless) Helicopter Rotor Systems*" [Ref. 9]. In the report the helicopter is designated as "H2" and its characteristics are shown in Appendix D.

Figure 5 shows the output menu that appears after *blade.m* has finished the required calculations. There are four view options from which the user may choose. The first option (1) shows the steady and first ten harmonic responses of the rotor blade. They are presented in eleven figure WINDOWS®, the first WINDOW® being the steady

```
    CHOOSE WHICH OUTPUT OPTION YOU WOULD LIKE

 1. View the steady and first ten harmonic responses

 2. View a mesh plot of the flatwise Shear, Moment, Slope and Deflection
      at all azimuth positions

 3. View the flatwise Shear, Moment, Slope and Shear at a specific
      azimuth position

 4. View the stiffness (EI) and weight distribution

 0. Exit

*** FOR A PRINTOUT CHOOSE THE "File" OPTION IN THE DESIRED GRAPH WINDOW ***

    >>
```

Figure 5. Output Option Menu

response, the second WINDOW® is the first harmonic response, and so on. For each harmonic blade response there are four subplots which show: blade flapwise shear, moment, blade slope, and deflection. Results for the steady (or zeroth) and first ten harmonic responses of the rotor blade are shown in Figure 6. If option two (2) is selected, the user will see a mesh plot of the total response of the rotor blade over one

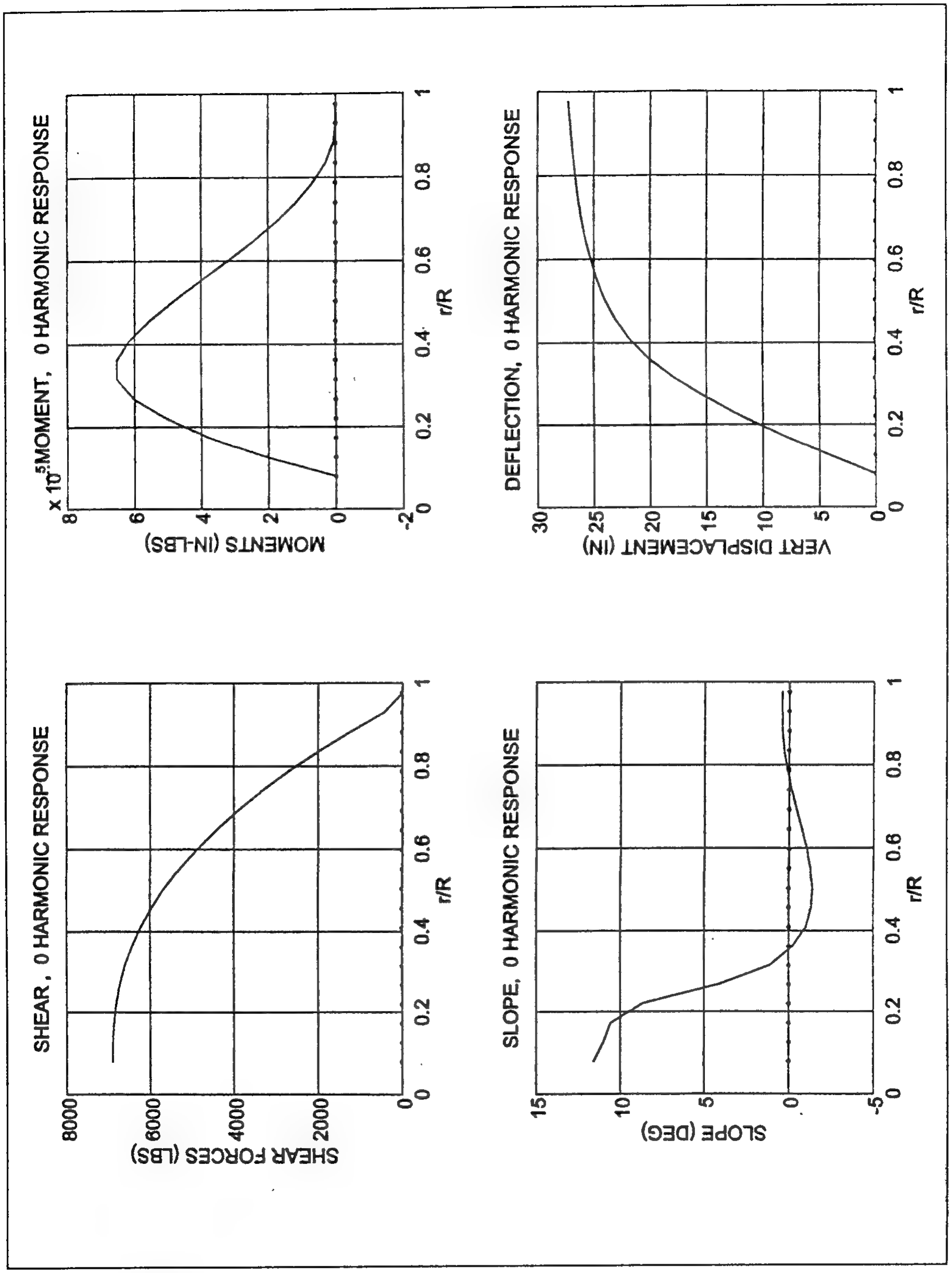

Figure 6. Steady (Zeroth) Response of Sample Articulated Rotor Blade "H2"

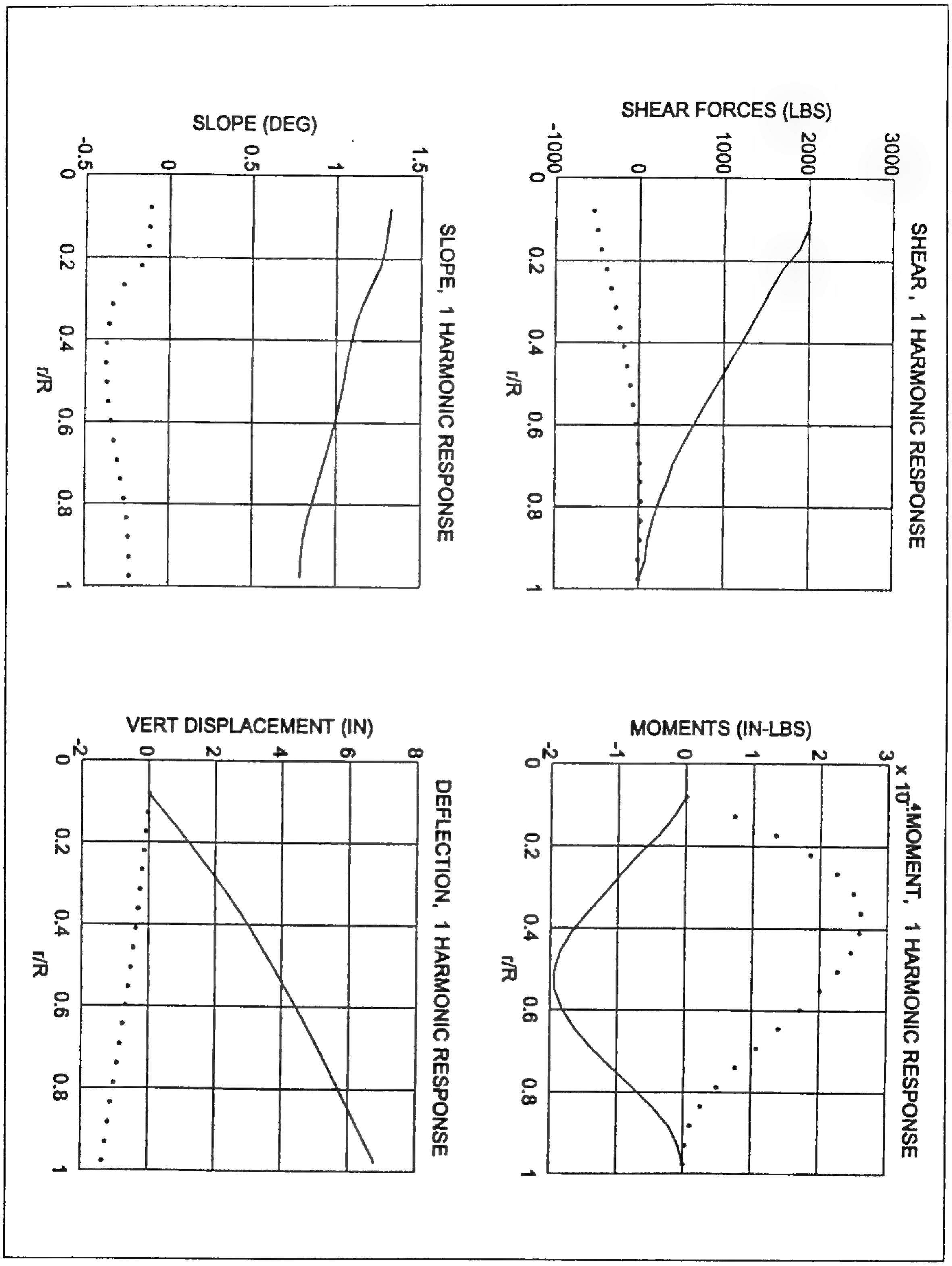

Figure 6 (Cont.) . First Harmonic Response of Sample Articulated Rotor Blade "H2"

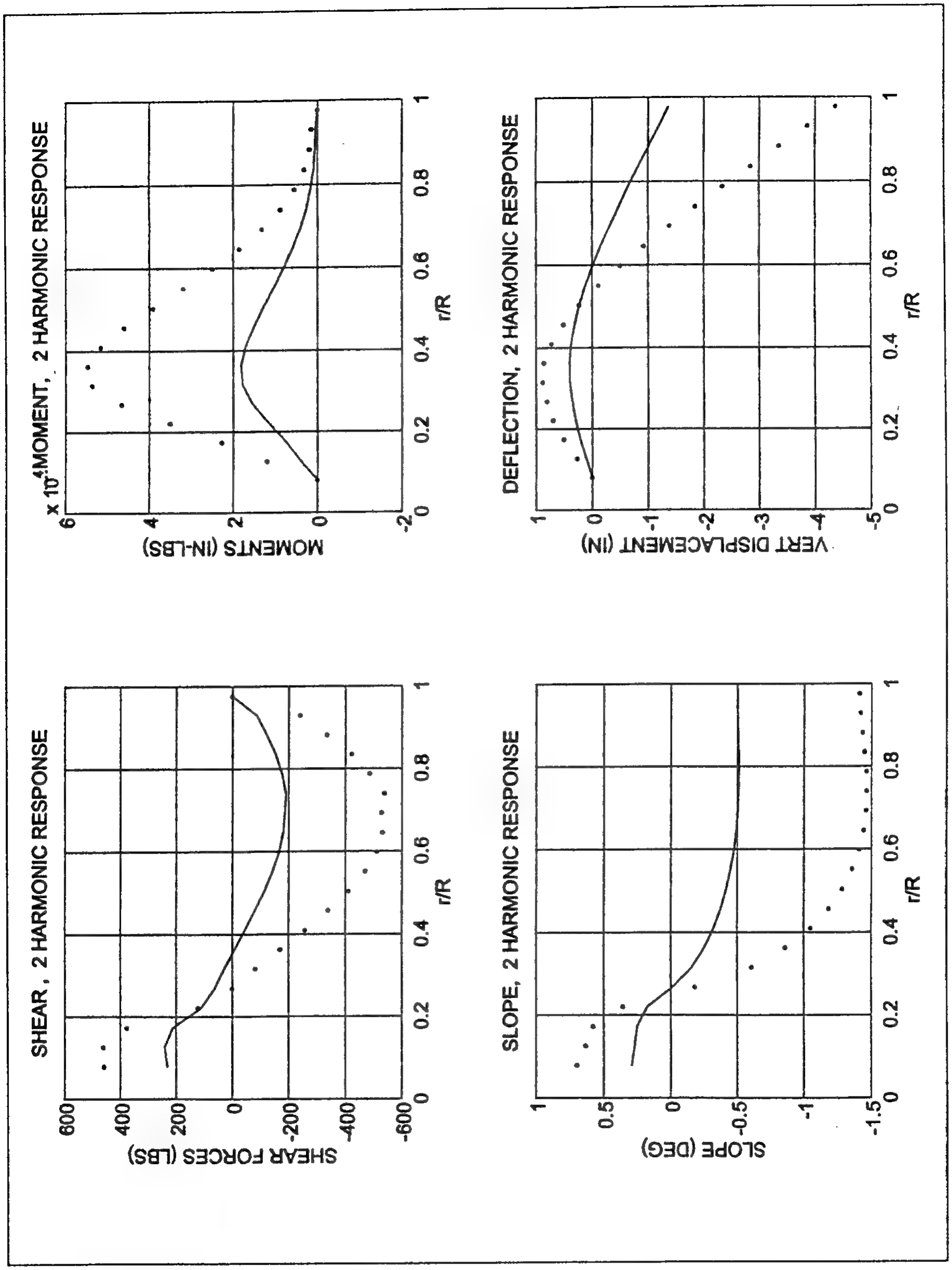

Figure 6 (Cont.) . Second Harmonic Response of Sample Articulated Rotor Blade "H2"

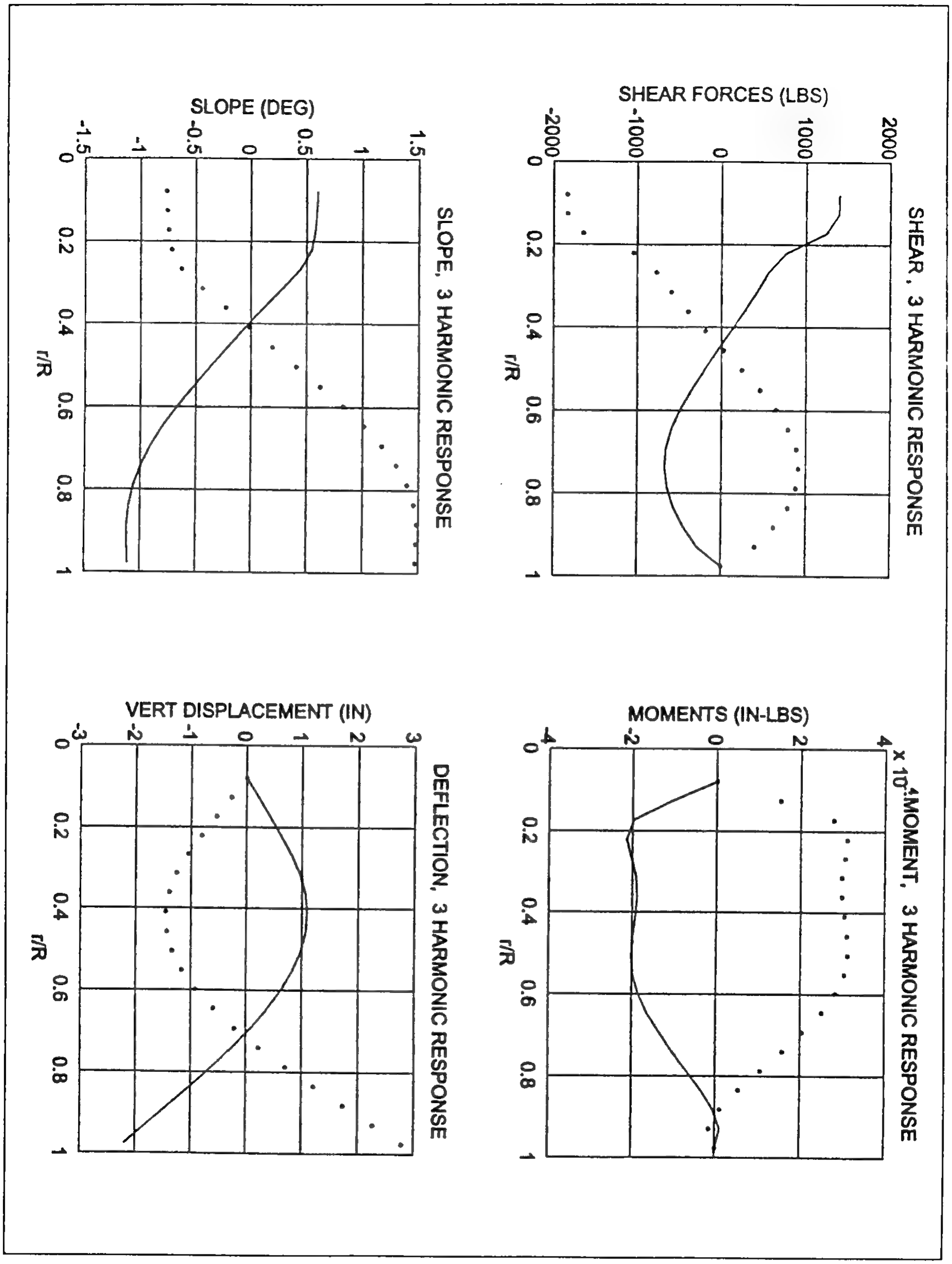

Figure 6 (Cont.) . Third Harmonic Response of Sample Articulated Rotor Blade "H2"

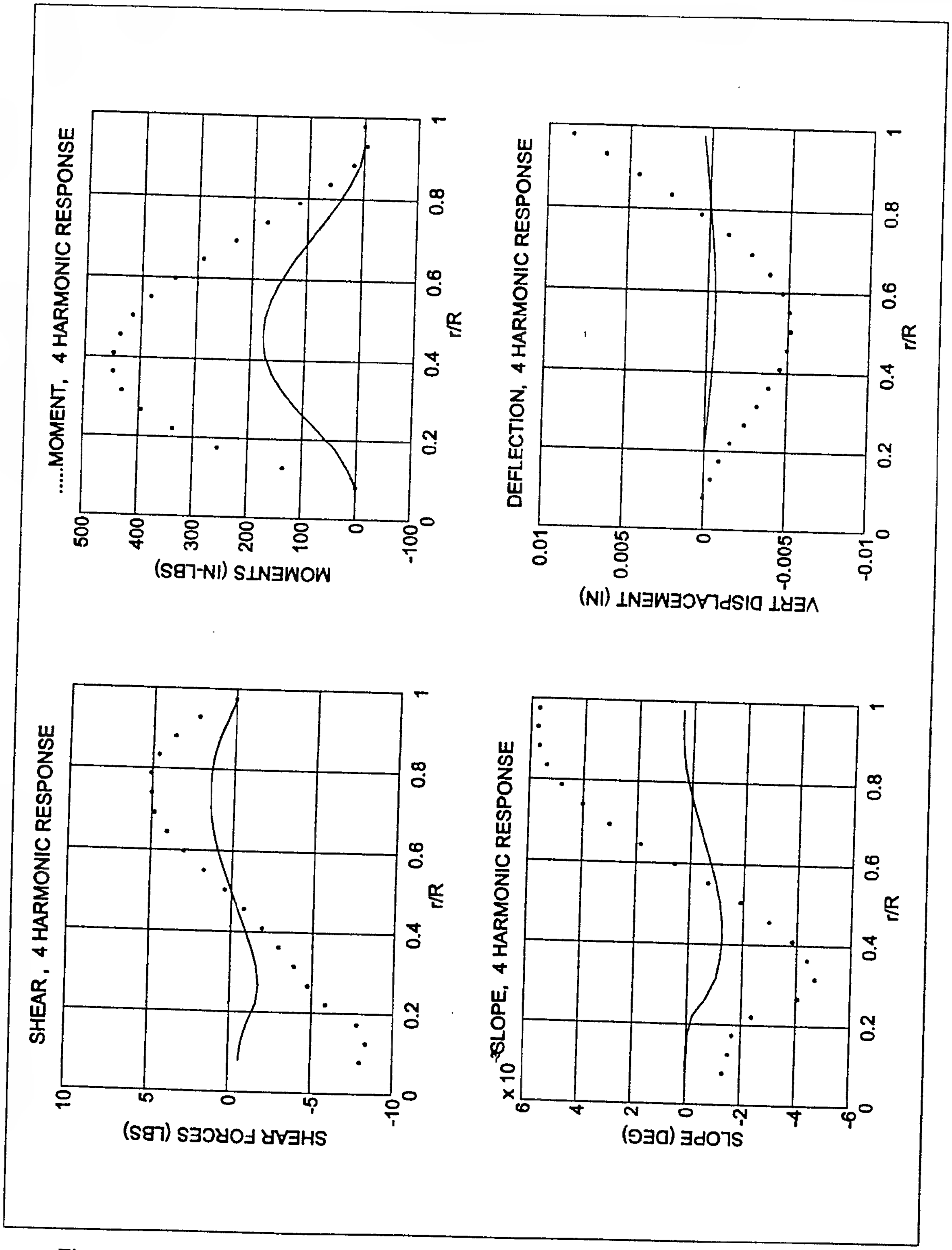

Figure 6 (Cont.). Fourth Harmonic Response of Sample Articulated Rotor Blade "H2"

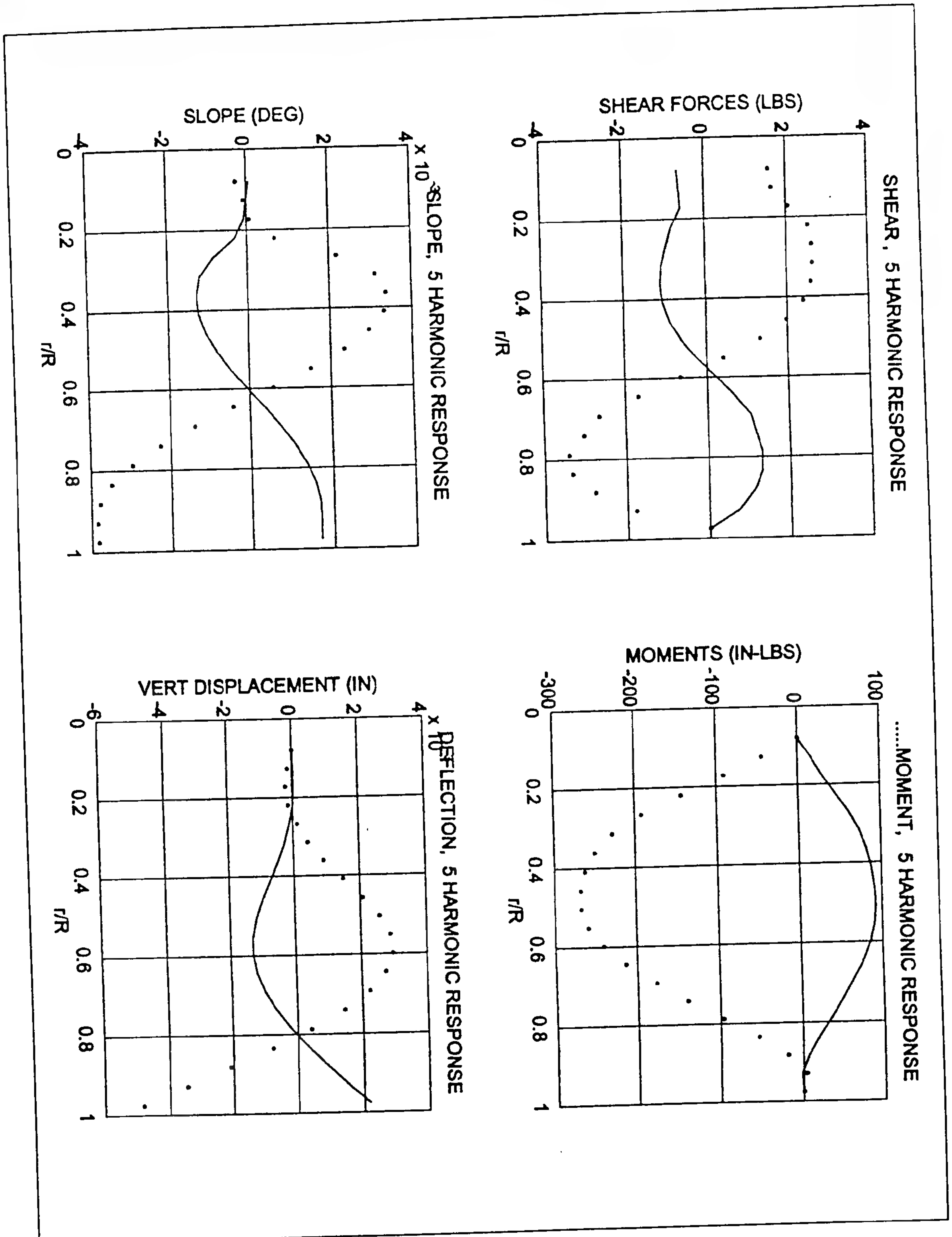

Figure 6 (Cont.). Fifth Harmonic Response of Sample Articulated Rotor Blade "H2"

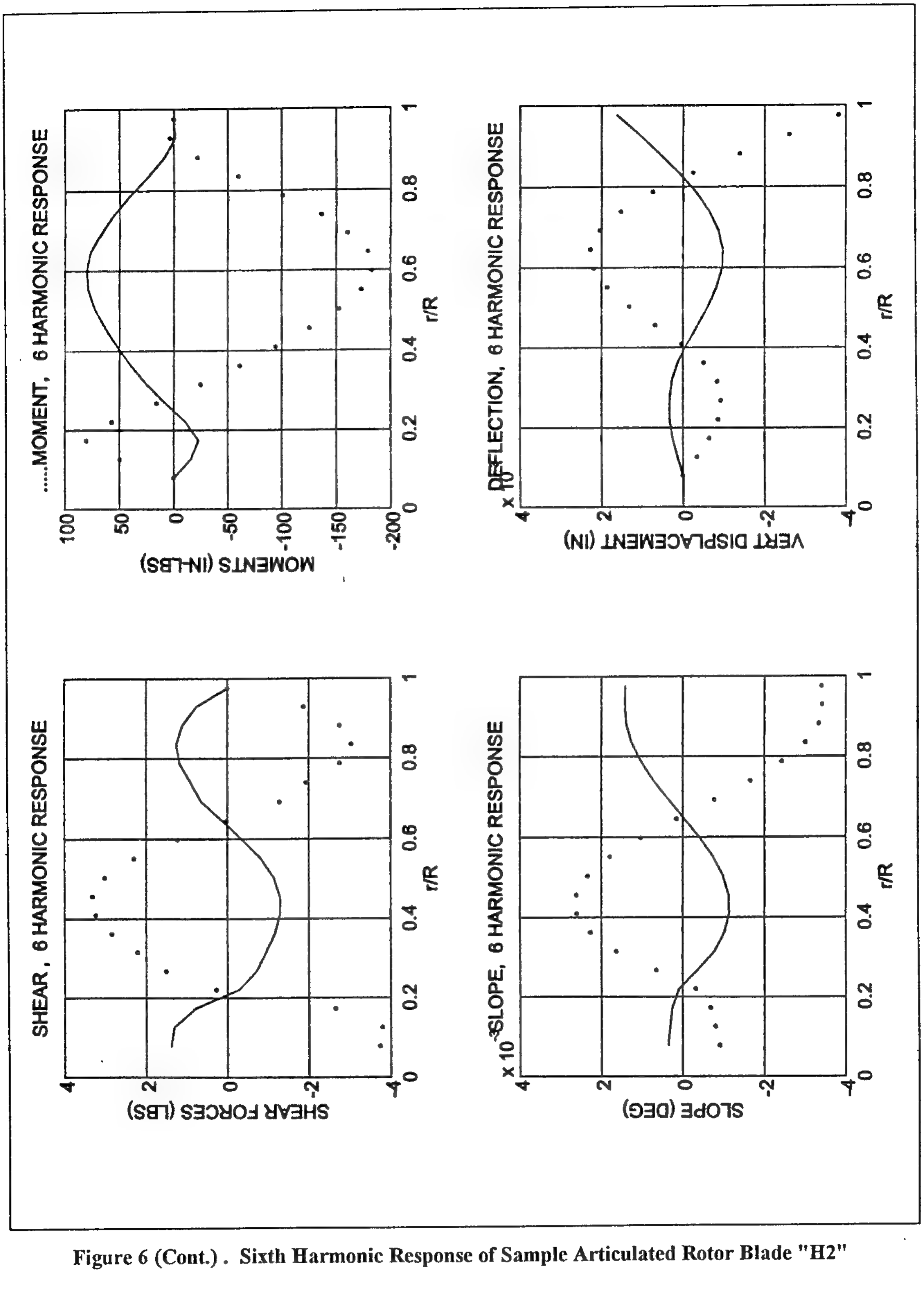

Figure 6 (Cont.) . Sixth Harmonic Response of Sample Articulated Rotor Blade "H2"

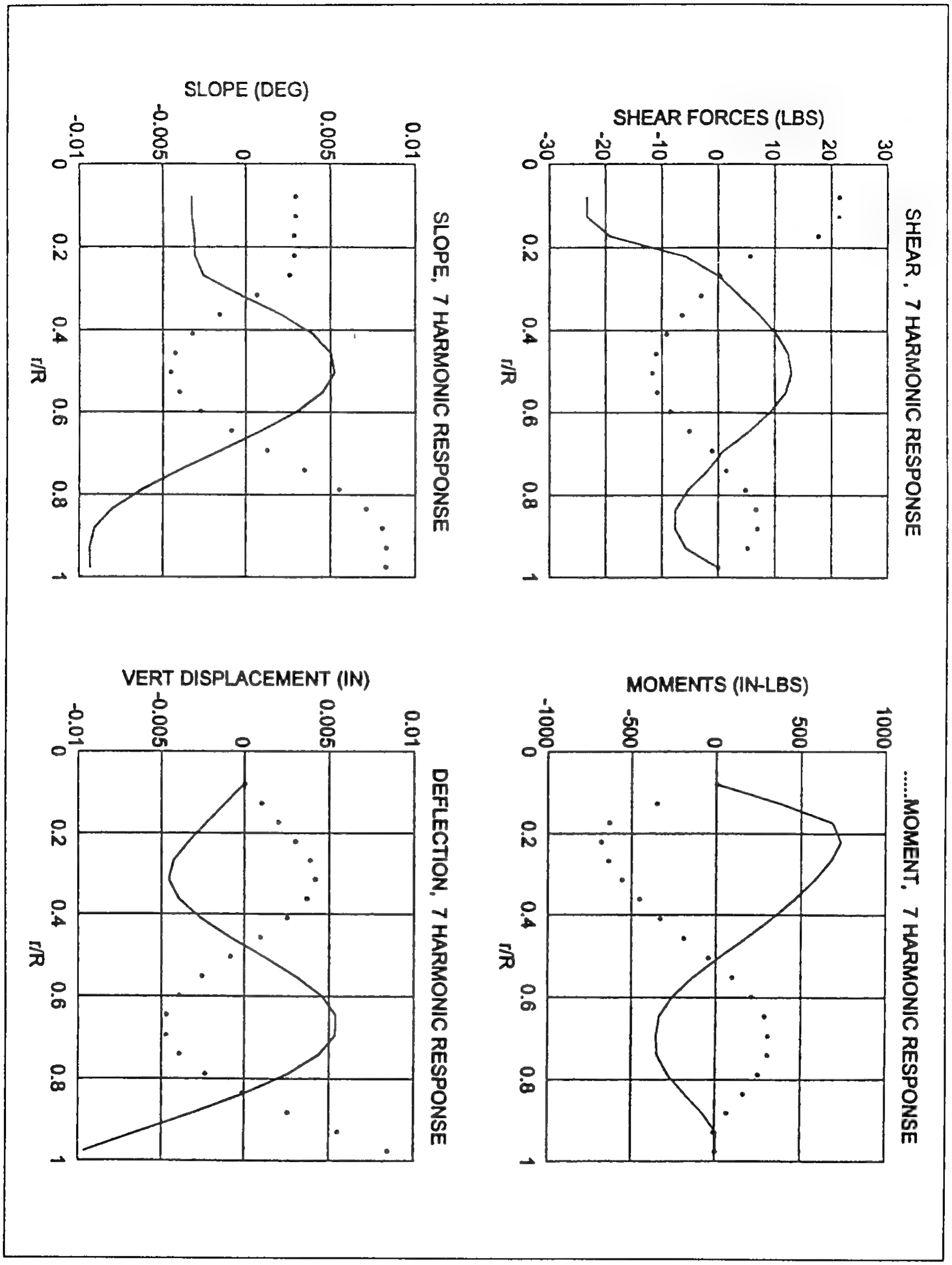

Figure 6 (Cont.) . Seventh Harmonic Response of Sample Articulated Rotor Blade "H2"

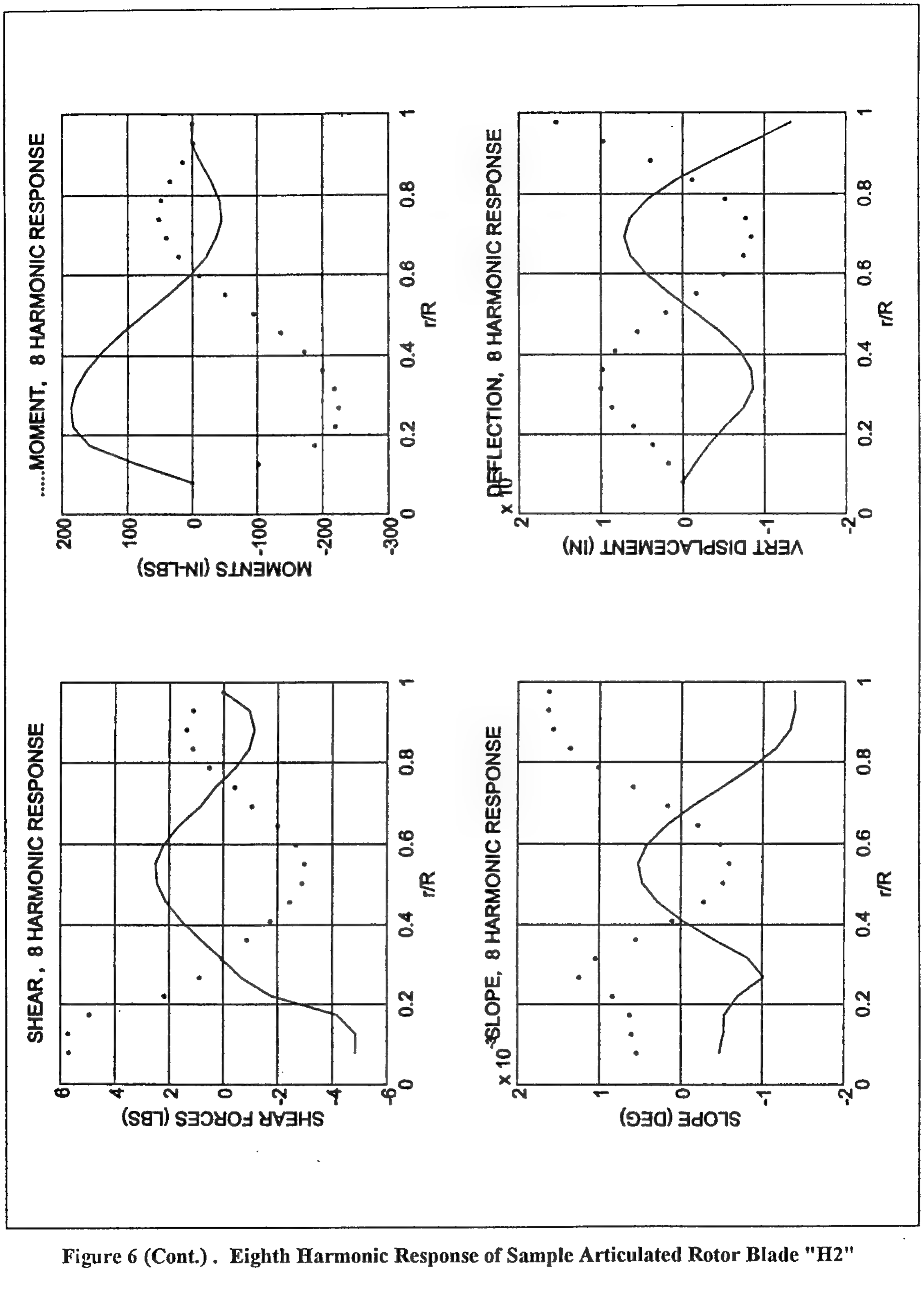

Figure 6 (Cont.) . Eighth Harmonic Response of Sample Articulated Rotor Blade "H2"

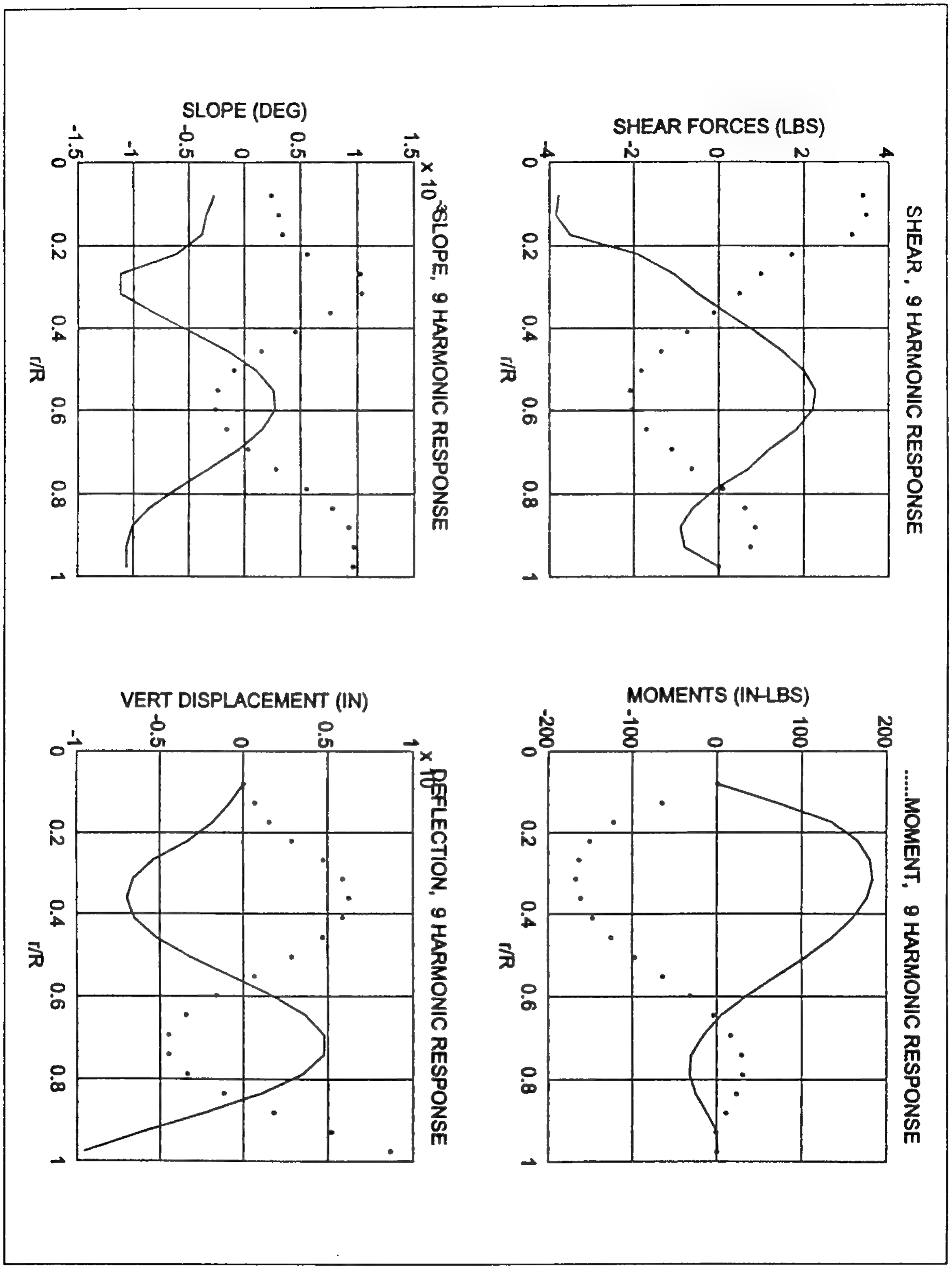

Figure 6 (Cont.) . Ninth Harmonic Response of Sample Articulated Rotor Blade "H2"

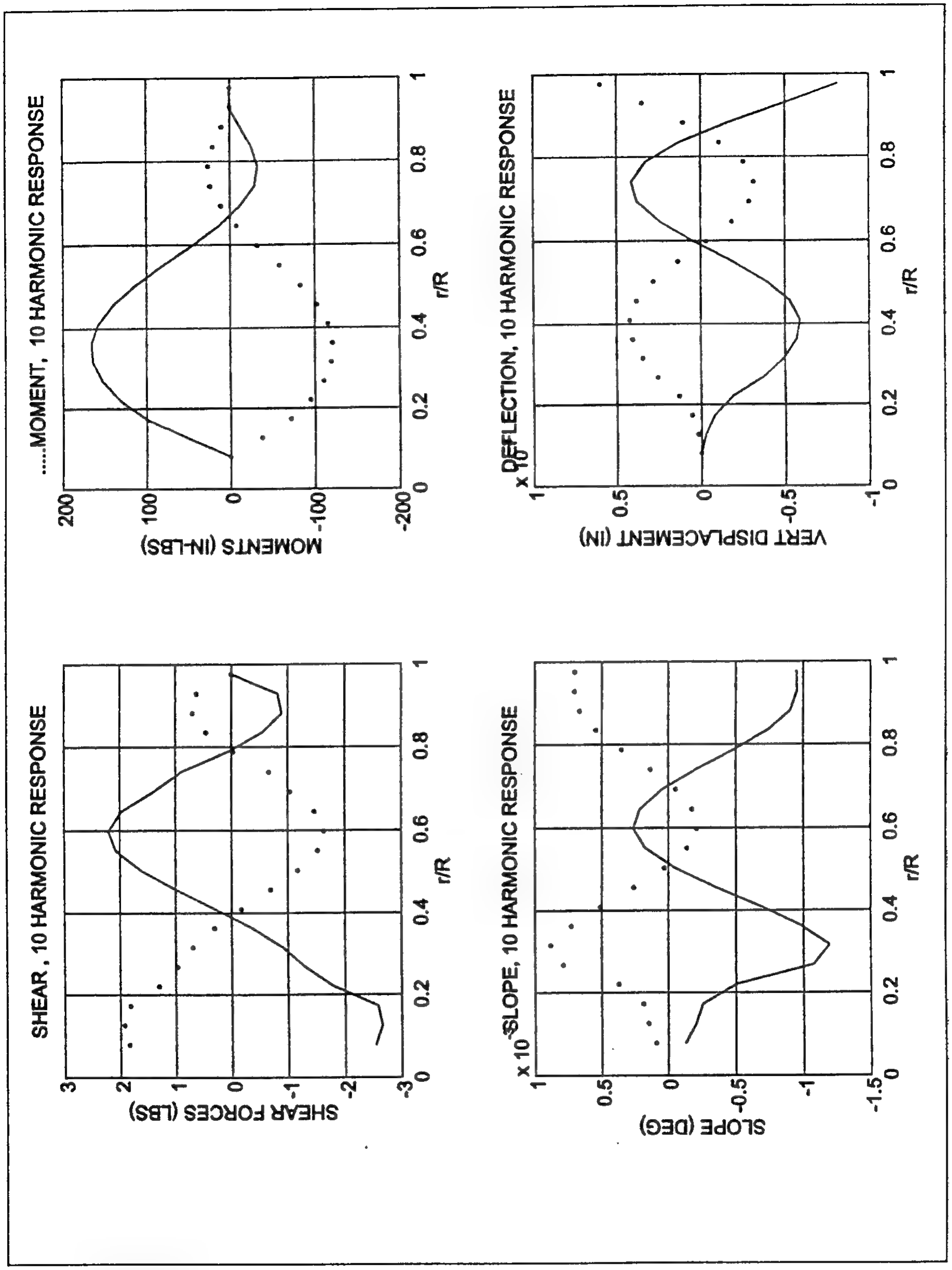

Figure 6 (Cont.) . Tenth Harmonic Response of Sample Articulated Rotor Blade "H2"

full revolution for the flatwise shear, moment, slope and deflection (shown in Figure 7). Option three (3), shown in Figure 8, allows the user to choose a specific angle to view the rotor blade's total response. Option four (4), shown in Figure 9, allows viewing of a graphical representation of the stiffness modulus and weight distribution.

In Figure 6 we see the steady (zeroth) response and the first ten harmonic responses of the sample articulated blade. We note that the dotted lines represent the imaginary component of the harmonic responses. Consistent with complex notation, the "imaginary" component of the response leads the "real" component by a 90° phase at the frequency designated. From the steady (zeroth) response it can be seen that the displacement represents the steady state "coning angle" of the articulated rotor system and since there is no phase dependency it contains no imaginary component. The steady and harmonic responses also confirm that all the appropriate boundary conditions are obeyed, namely: that moments at the blade hinge and tip are zero, deflection at the blade hinge is zero, and vertical shear at the tip is zero. The characteristic mode shapes of the helicopter blade can be seen to develop as the frequency harmonic increases up to the third mode shape, typical of the blade's tenth harmonic response.

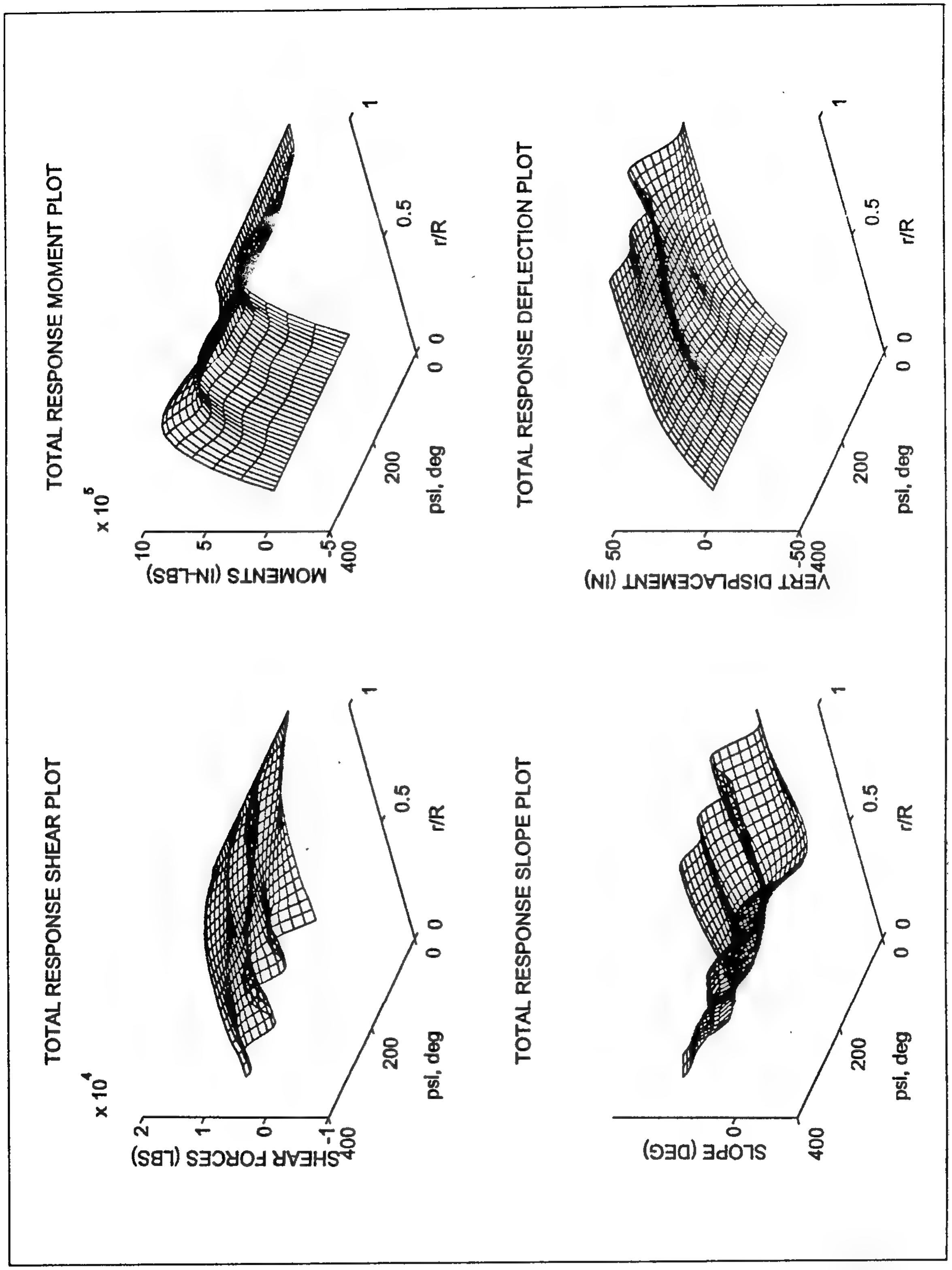

Figure 7. Mesh Plot of Total Response of Articulated Rotor Blade "H2" Over One Full Revolution

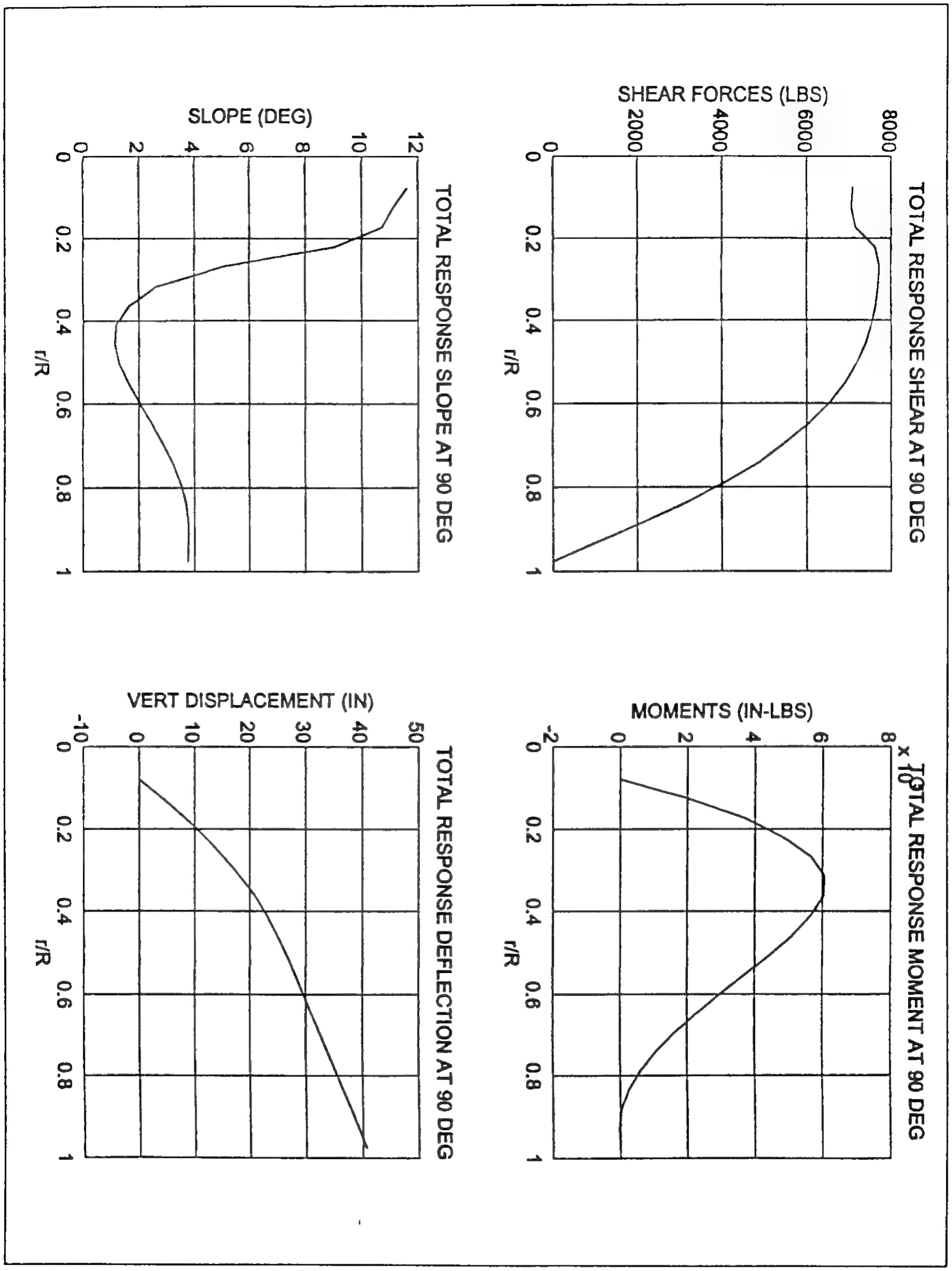

Figure 8. Total Response of Articulated Rotor Blade "H2" at a Chosen Azimuth Angle of 90 Deg.

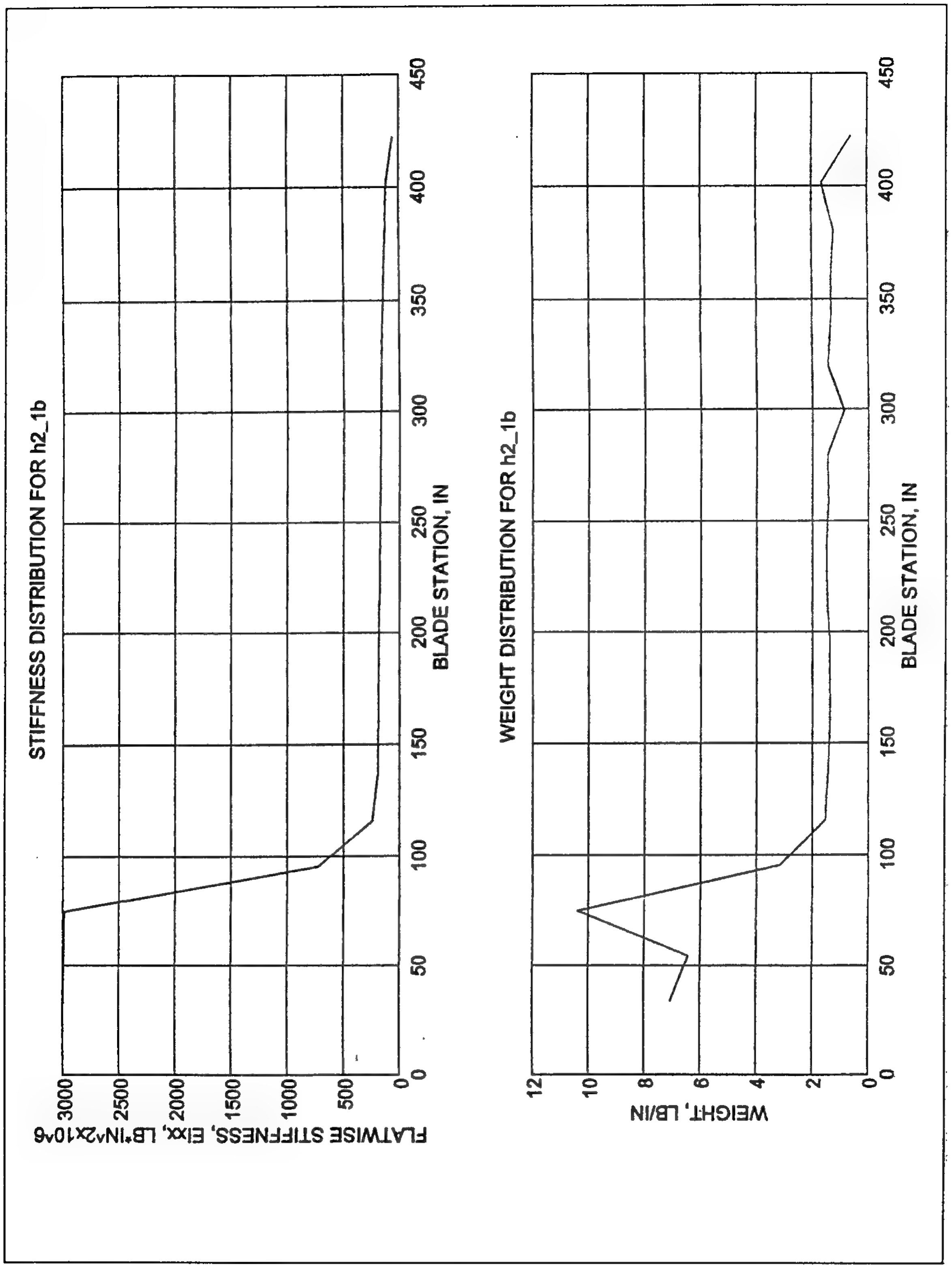

Figure 9. Graph of the Stiffness Modulus and Weight Dist. for Articulated Rotor Blade "H2"

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The modified Myklestad-Prohl method for rotating beams has been described and used in approximating solutions to aerodynamically forced vibrations of helicopter rotor blades. This type of dynamic response analysis has made it possible to investigate the dynamic properties of a helicopter rotor blade of varying material and sectional properties. JANRAD, as a preliminary design tool has been expanded to better meet the needs of a helicopter design course. In this respect, Matlab® has shown itself to be the programming language of choice due to its availability, versatility in matrix manipulation, and ease of use. Although sample helicopter data is only available for the fully coupled blade, results show good agreement with results given in Ref. 9. Individual mode shape responses also show good qualitative agreement to harmonic excitations.

B. RECOMMENDATIONS

In this thesis we have looked at the flatwise uncoupled response of a helicopter rotor blade to the effects of individual harmonics of airloads. The analysis includes such effects as aerodynamic damping, and centrifugal forces. Articulated blade root end conditions have been treated in this thesis. An immediate improvement to this program code would be to add hingeless and teetering rotor blade root construction and also include provisions for blade lead-lag or inplane response with lead-lag dampers accounted

for in the boundary conditions. This would require adding chordwise dynamics with twist coupling into the Myklestad equations.

The Rotor Performance Analysis and Rotor Dynamic Analysis are both based on a more extensive aeroelastic analysis that is described in detail in Ref. 3. JANRAD so far has looked at the main cause of the helicopter's vibrations, i.e. the rotor system. For a truly comprehensive investigation of the effects of rotor response, one should "marry" the rotor blades to the helicopter fuselage. As a follow-on thesis project the code may be expanded to include rotor blade-fuselage coupling through impedance matching, this is discussed in detail in Ref. 3. Having done this, the forced response of the helicopter fuselage may be calculated and we would have available an excellent tool to be applied in both preliminary and detailed design of a helicopter. It could also be used for research studies of rotor/airframe systems.

APPENDIX A: DYNAM.M

```
% DYNAM.M
% JANRAD: NPS Helicopter Preliminary Design Program
% Rotor Blade Dynamic Response Routines
% Written by Lt Juan D. Cuesta
% September 1994

% This program was designed as an interactive preliminary design tool
% for rotor blade dynamic analysis and design of either an articulated
% or hingeless rotor blade system. The program provides the shear,
% moment, slope angle, and deflection of the flatwise response at any
% point along the length of a rotor blade to the steady and first ten
% harmonic aerodynamic loads. This data can then plotted at various
% azimuth blade positions. The input of variables follows the same
% format as written by Majs Bob Nicholson, Jr. and Walter Wirth, Jr.
% for JANRAD.

% Variable List for Dynam.m, Blade.m, output.m

% a          lift curve slope
% alphaFn    elemental force column matrix
% az         azimuth position angle
% bc         root boundary condition
% cblade2    blade chord at radial segment, from tip to root
% Cn         flatwise aerodynamic damping on blade element
% delr       blade radial segment width, starting from blade hinge
% dfn        imaginary component of harmonic thrust terms
% dFn        real component of harmonic thrust terms
% dFo        steady state thrust terms
% dT         JANRAD Performance routine thrust output
% EI         elemental bending modulus
% En         elemental modulus of elasticity
% filename1  .mat file with janrad data
% filename3  .mat file which contains blade data
% Fn         distribution of thrust airloads from tip to root
% Ibn        distribution of moment of inertia of blade elements
% lsn        length of blade segment
% mn         distribution of blade mass
% Mn         flatwise moment for blade element
% naz        number of azimuth sectors
% nbe        number of blade elements
% omega      rotor rotational velocity
% omegae     excitation frequency
% Pn         running transfer matrix along blade length
% psi        azimuth angle
% rho        ambient air density
% rn         radius, rotor blade radial segment, from blade hinge
% Sn         flatwise shear for blade element
% Thetan     flatwise slope of blade element
% tip_art_bc articulated blade tip slope and deflection bound. cond.
% tip_rig_bd hingeless blade tip slope and deflection bound. cond.
% Tn         elemental radial tension
% Un         Transfer matrix between adjacent blade elements
```

```
% view       option variable for viewing choice
% Wn         distribution of blade weight
% X,Y,Yout   output data to generate graphs
% Yn         flatwise deflection of blade element
% Zn         rotor blade element state vector
% Zroot      rotor blade root state vector
% Ztip       rotor blade tip state vector

flag=1;
flag=exist('filename1');
if flag == 0,
  disp(' ')
  disp('   *** You must run Rotor Performance Analysis first ***')
  disp('              *** Press Any Key to Continue ***')
  disp(' ')
  pause
else,
  eval (['load ',filename1 '_p'])
  eval (['load ',filename1])
  clc
disp(' ')
disp(' ')
disp('       ***   ROTOR BLADE DYNAMIC ANALYSIS ROUTINE   ***')
disp(' ')
disp(' ')

disp('   Do you want to edit an existing blade file or create a new one?')

  answer3=input('   1. edit existing file    2. create new file    >>');

  if answer3==1,
    clc
disp(' ')
disp(' ')
disp('                *** LOAD INSTRUCTIONS *** ')
disp(' ')
disp('         A. Input the name of the rotor blade data file to edit.')
disp('         B. The file was saved in your previous session')
disp('           with a ".mat" extension.')
disp('         C. Do not include the extension or quotations.')
disp(' ')
disp('         ex: blade1')
    flag=0;
    while flag < 1
      filename3=input('        Enter the name of Blade data input file: ','s');
      eval(['flag=exist(''',filename3,'.mat'');'])
      if flag < 1,
disp(' ')
disp('         The file does not exist, try again or < Ctrl-C >')
disp('         to exit program.')
      end
    end
    eval(['load ',filename3])
   check4=1;
   while check4 > 0,
    clc
```

```
disp(' ')
disp('           ***BLADE DYNAMICS EDIT MENU ***')
disp(' ')
disp('         1. root boundary condition')
disp('         2. blade material properties')
disp('         3. weight distribution')
disp(' ')
disp(' ')
disp('        *** ALL OTHER BLADE INFORMATION IS ENTERED IN MAIN JANRAD MENU ***')
disp(' ')
disp('         0. NO CHANGES')

   choice1=input('        Input the parameter to change: ');
   if isempty(choice1),
    choice1=0;
   end
   if choice1==1,
    clc
    disp(' ')
    disp('Root Boundary Condition')
    bc
    tmp=bc;
    flag=1;
    while flag > 0
     bc=input('Root Boundary Condition  1. Articulated  2. Hingeless   >> ');
     if isempty(bc),
      bc=tmp;
     end
     if bc==1,
      flag=0;
     elseif bc==2,
      flag=0;
     else
      disp(' ')
      disp(' *** Enter a 1 or 2 ***')
     end
    end
   end
   if choice1==2,
    clc
    disp(' ')
    disp('1. variable elasticity, E, and variable moment of inertia, Ixx')
    disp('2. variable stiffness, EI')
    option=input('Choose 1. OR 2.  >> ');
    if option==1,
     clc
     E=En/1e6
     tmp=En;
     disp('1) Enter as a row vector starting from the')
     disp('    tip and ending with the root; ex: "[18 18.1 .... 21]"')
     disp(' ')
     fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
     disp(' ')
     disp('3) Enter the modulus of ELASTICITY, E, distribution (lbs/in^2 x 10^6): ')
     En=input(' >>').*1e6;
     if isempty(En),
```

```
    En=tmp;
   end
   while (length(En)~=nbe),
    disp('1) Enter as a row vector starting from the ')
    disp('    tip and ending with the root; ex: "[18 18.1 .... 21]"')
    disp(' ')
    fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
    disp(' ')
    disp('3) Enter the modulus of ELASTICITY, E, distribution (lbs/in^2 x 10^6): ')
    En=input(' >>').*1e6;
   end
   clc
   Ibn
   tmp=Ibn;
   disp('1) Enter as a row vector starting from the ')
   disp('    tip and ending with the root; ex: "[3.9 4.09 .... 15.1]"')
   disp(' ')
   fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
   disp(' ')
   disp('3) Enter blade flapping moment of INERTIA  Ixx, distribution (in^4): ')
   Ibn=input(' >>');
   if isempty(Ibn),
    Ibn=tmp;
   end
   while (length(Ibn)~=nbe),
    disp('1) Enter as a row vector starting from the ')
    disp('    tip and ending with the root; ex: "[3.9 4.09 .... 15.1]"')
    disp(' ')
    fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
    disp(' ')
    disp('3) Enter blade flapping moment of INERTIA, Ixx, distribution (in^4): ')
    Ibn=input(' >>');
   end
   if exist('EI')==1
    clear EI
   end
 end
if option==2,
  clc
  EI
  tmp=EI
  disp('1) Enter as a row vector starting from the ')
  disp('    tip and ending with the root; ex: "[70.2 73.62 .... 271.8]"')
  disp(' ')
  fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
  disp(' ')
  disp('3) Enter the bending/stiffness modulus, EI, distribution (lbs in^2 x 10^6): ')
  EI=input(' >>').*1e6;
   if isempty(EI),
    EI=tmp;
   end
   while (length(EI)~=nbe),
    disp('1) Enter as a row vector starting from the ')
    disp('    tip and ending with the root; ex: "[70.2 73.62 .... 271.8]"')
    disp(' ')
    fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
```

```
        disp(' ')
        disp('3) Enter the bending stiffness, EI, distribution (lbs in^2 x 10^6): ')
        EI=input(' >>').*1e6;
      end
      if exist('En')
        clear En Ibn
      end
    end
  end
  if choice1==3,
    clc
    Wn
    tmp=Wn;
    disp(' ')
    fprintf('1) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
    disp(' ')
    disp('2) Enter as a row vector starting from the ')
    disp('    tip and ending with the root; ex: "[9.86 9.95 .... 11.96]"')
    disp(' ')
    disp('3) THE TOTAL WEIGHT MUST BE GREATER THAN THE AERODYNAMIC')
    fprintf('    PORTION OF THE BLADE: %6.2f\n',wblade)
    disp(' ')
    disp('weight distribution (lbs/segment): ')
    Wn=input(' >>');
    if isempty(Wn),
      Wn=tmp;
    end
    while (length(Wn)~=nbe),
      disp(' ')
      fprintf('1) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
      disp(' ')
      disp('2) Enter as a row vector starting from the ')
      disp('    tip and ending with the root; ex: "[9.86 9.95 .... 11.96]"')
      disp(' ')
      disp('3) THE TOTAL WEIGHT MUST BE GREATER THAN THE AERODYNAMIC')
      fprintf('    PORTION OF THE BLADE: %6.2f\n',wblade)
      disp(' ')
      disp('weight distribution (lbs/segment): ')
      Wn=input(' >>');
    end
  end
  if choice1==0,
    clc
disp(' ')
disp(' ')
disp('                    *** SAVE INSTRUCTIONS ***')
disp(' ')
disp('          A. Save the new data to a specified file name.')
disp('          B. Do not use an extension or quotations.')
disp('          C. Use letter/number combinations of 6 characters or less.')
disp('          D. The file will be saved with a ".mat" extension.')
disp(' ')
disp('          ex: blade1')
disp(' ')
disp('          E. If you made no changes, press < Enter >, the file will')
disp('             be saved with the original name.')
```

```
        flag=1;
        while flag > 0
          filename0=filename3;
          filename3=input('        save file as: ','s');
          if isempty(filename3),
            filename3=filename0;
          end
          clear filename0
          if length(filename1) > 6,
            disp(' ')
            disp('         use 6 characters or less')
            flag=1;
          else
            flag=0;
          end
          eval(['save ',filename3])
         check4=0
        end
      end
    end
   end
% Creating a new file
    if answer3==2,
      change=1;
      while change > 0,
        clc
        bc=input('Root Boundary Condition  1. Articulated  2. Hingeless   >> ');
        while isempty(bc),
          disp(' ')
          disp('You must enter a numerical value')
          bc=input('Root Boundary Condition  1. Articulated  2. Hingeless   >> ');
        end
        disp(' ')
        disp('     Do you want to enter:')
        disp(' ')
        disp('     1. elasticity, E, AND flapping moment distribution, Ixx')
        disp('     OR')
        disp('     2. just the bending stiffness, EI, distribution.')
        clear option1
        option1=input('Enter 1 or 2   >> ');
        while isempty(option1),
        option1=input('Enter 1 or 2   >> ');
      end
      clc
      if option1==1,
        disp(' ')
        disp('1) Enter as a row vector starting from the ')
        disp('    tip and ending with the root; ex: "[18 18.1 .... 21]"')
        disp(' ')
        fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
        disp(' ')
        disp('3) Enter the modulus of ELASTICITY, E, distribution (lbs/in^2 x 10^6):')
        En=input('>> ').*1e6;
        while (length(En)~=nbe),
          disp(' ')
          fprintf('1) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
```

```
  disp(' ')
  disp('2) Enter the modulus of ELASTICITY, E, distribution (lbs/in^2 x 10^6):')
  En=input('>> ').*1e6;
 end
 disp(' ')
 disp('1) Enter as a row vector starting from the ')
 disp('    tip and ending with the root; ex: "[3.9 4.09 .... 15.1]"')
 disp(' ')
 fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
 disp(' ')
 disp('3) Enter blade flapping moment of INERTIA, Ixx, distribution (in^4): ')
 Ibn=input('>> ');
 while (length(Ibn)~=nbe),
  disp(' ')
  fprintf('1) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
  disp(' ')
  disp('2) Enter blade flapping moment of INERTIA, Ixx, distribution (in^4): ')
  Ibn=input('>> ');
 end
end
if option1==2,
 disp(' ')
 disp('1) Enter as a row vector starting from the ')
 disp('    tip and ending with the root; ex: "[70.2 73.62 .... 271.8]"')
 disp(' ')
 fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
 disp(' ')
 disp('3) Enter the bending stiffness, EI, distribution (lbs in^2 x 10^6): ')
 EI=input('>> ').*1e6;
 while (length(EI)~=nbe),
  disp(' ')
  fprintf('1) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
  disp('2) Enter the bending stiffness, EI, distribution (lbs in^2 x 10^6): ')
  EI=input('>> ').*1e6;
 end
end
disp(' ')
disp('1) Enter as a row vector starting from the ')
disp('    tip and ending with the root; ex: "[9.86 9.95 .... 11.96]"')
disp(' ')
fprintf('2) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
disp(' ')
disp('3) THE TOTAL WEIGHT MUST BE GREATER THAN THE AERODYNAMIC')
fprintf('    PORTION OF THE BLADE: %6.2f\n',wblade)
disp(' ')
disp('Enter weight distribution (lbs/segment)')
Wn=input('>> ');
while (length(Wn)~=nbe),
 disp(' ')
 fprintf('1) YOU MUST ENTER%3.0f ELEMENTS\n',nbe)
 disp('2) Enter weight distribution (lbs/segment)')
 Wn=input('>> ');
end
clc
disp(' ')
disp(' ')
```

```
    disp('          *** DATA ENTRY COMPLETE ***')
    disp('           PLEASE REVIEW YOUR DATA')
    disp(' ')
    disp('          PRESS ANY KEY TO CONTINUE')
    pause
    if option1==1,
     E=En/1e6
     Ibn
    end
    if option1==2,
     EI
    end
    Wn
    disp('Do you wish to make any changes?')
    change=input('0. No   1. Yes  >>');
   end

   clc
disp(' ')
disp(' ')
disp('                    *** SAVE INSTRUCTIONS ***')
disp(' ')
disp('         A. Save the data to a specified file name.')
disp('         B. Do not use an extension or quotations.')
disp('         C. Use letter/number combinations of 6 characters or less.')
disp('         D. The file will be saved with a ".mat" extension.')
disp(' ')
disp('         ex: blade1')
disp(' ')
disp('         E.  If you do not enter a name, the default is "blade1" ')
   flag=1;
   while flag > 0
    filename3=input('        save file as: ','s');
    if isempty(filename3),
     filename3='blade1';
    end
    if length(filename3) > 6,
     disp(' ')
     disp('         use 6 characters or less')
     flag=1;
    else
     flag=0;
    end
   end
   if exist('EI')==0,
    eval(['save ',filename3,' bc En Ibn Wn'])
   end
   if exist('En')==0,
    eval(['save ',filename3,' bc EI Wn'])
   end
  end
  blade
end
```

APPENDIX B: BLADE.M

```
% forced response program for flapwise blade dynamics
% utilizing the Myklestad-Prohl method

clc
disp(' ')
disp(' ')
disp('              *** UTILIZING THE MYKLESTAD-PROHL METHOD TO SOLVE ***')
disp('                   *** FOR FORCED BLADE DYNAMIC RESPONSE ***')

delr=(R-e)/nbe;

for k=nbe:-1:1,
  rn((nbe+1)-k)=e+k*delr-delr/2;
end
rn=(rn*12);

Wn=Wn/12; %lb/in

mn=Wn/(32.174); %lbm/in

disp(' ')
disp('              *** CALCULATING THE STEADY AND TEN HARMONIC ***')
disp('                  *** DIFFERENTIAL THRUST ELEMENTS ***')

%average the first and last two columns to reduce dT by 2 columns
%reverse columns so radius goes from tip to root
Fn=fliplr([(dT(:,1)+dT(:,2))./2, dT(:,3:(nbe-1)), (dT(:,nbe)+dT(:,nbe+1))./2]);

%calculate harmonics:
%setting up rows of harmonics and columns of blade stations

%Calculate steady state term
for j=1:nbe-1,
  dFo(j)=(sum(Fn(:,j)))/naz;
end

%Calculate dFn (sin term) harmonic matrix
for i=1:10,
  for j=1:nbe-1,
    dFn(i,j)=0;
    for k=1:naz,
      Fcum=Fn(k,j)*sin(i*psi(k)/57.3);
      dFn(i,j)=dFn(i,j)+Fcum;
    end
    dFn(i,j)=dFn(i,j)/(naz/2);
  end
end

%now calculate dfn (cos term) harmonic matrix
for i=1:10,
  for j=1:nbe-1,
    dfn(i,j)=0;
```

```
    for k=1:naz,
      fcum=Fn(k,j)*cos(i*psi(k)/57.3);
      dfn(i,j)=dfn(i,j)+fcum;
    end
    dfn(i,j)=dfn(i,j)/(naz/2);
  end
end

% to iterate we switch the n's in the myklestad eqns (ref p15,
% "an intro to helo dynamics", E. R. Wood) to k+1's
% in the vector indices and vice versa
% iterations will always be from tip to root

disp(' ')
disp('               *** CALCULATING THE AERODYNAMIC DAMPING ***')

% Calculating the flatwise aerodynamic damping, Cn, on each blade element.
cblade2=fliplr(cblade);
for k=1:nbe-1,
Cn(k)=a*cblade2(k+1)*(rn(k)-rn(k+1))*.5*rho/(12)^4*omega*rn(k);
end
%Cn=zeros(1,23);

% the states are: Z=[Sn; Mn; Thetan; Yn]
% FIRST CALCULATE P of equation:
% [Z](at root) = [P]*[Z](at tip) + alphaF(at root)

disp(' ')
disp('*** CALCULATING THE TRANSFER MATRICES FOR THE STEADY AND 10 HARMONIC ***')
disp('          *** BLADE RESPONSES ALONG THE LENGTH OF THE BLADE ***')

% CALCULATING TENSION first, since it is independent
Tn(1)=mn(1)*omega^2*rn(1);

for k=1:nbe-1,
  Tn(k+1)=Tn(k) + mn(k+1)*omega^2*rn(k+1);
end

for i=0:10;

  omegae=omega*i;

  Pn=eye(4);
  for k=1:nbe-1,
    lsn=rn(k)-rn(k+1);
    if (exist('EI')==0),
      EI=En.*Ibn;
    end

    Un=[ 1+((mn(k+1)*omegae^2-sqrt(-1)*Cn(k)*omegae)*lsn^3)/(6*EI(k+1)),...
         ((mn(k+1)*omegae^2-sqrt(-1)*Cn(k)*omegae)*lsn^2)/(2*EI(k+1)),...
        -(mn(k+1)*omegae^2-sqrt(-1)*Cn(k)*omegae)*(Tn(k)*lsn^3/(6*EI(k+1))+lsn),...
            (mn(k+1)*omegae^2-sqrt(-1)*Cn(k)*omegae);

     lsn+(Tn(k)*lsn^3)/(6*EI(k+1)), 1+(Tn(k)*lsn^2)/(2*EI(k+1)),...
        -Tn(k)*(Tn(k)*lsn^3/(6*EI(k+1))+lsn), 0 ;
```

```
    -(lsn^2/(2*EI(k+1))), -(lsn/EI(k+1)), (1+Tn(k)*lsn^2/(2*EI(k+1))), 0 ;

    (lsn^3/(6*EI(k+1))), (lsn^2/(2*EI(k+1))), -(Tn(k)*lsn^3/(6*EI(k+1))+lsn), 1 ];

  Unew=Un;
  if k==1,
   Ucum=Unew;
  else
   U=[Ucum Unew];
   Ucum=U;
  end

  Pn=Un*Pn;

% tip BC for all cases:

  Sn(1)=0;
  Mn(1)=0;

  if i==0,
   Sn(k+1)=Sn(k)+dFo(k);
  else,
   Sn(k+1)=Sn(k)+dFn(i,k)+sqrt(-1)*dfn(i,k);
  end
  Mn(k+1)=Mn(k)+Sn(k)*lsn;
  Thetan(k+1)=-lsn*Mn(k)/EI(k+1)-lsn^2*Sn(k)/(2*EI(k+1));
  Yn(k+1)=-lsn^2*Mn(k)/(2*EI(k+1))-lsn^3/(3*EI(k+1));

  alphaFn(:,k)=[Sn(k+1); Mn(k+1); Thetan(k+1); Yn(k+1)];
 end
 P=Pn;

% SOLVING FOR Thetan, Yn (NON-ZERO BOUNDARY CONDITIONS) AT TIP

% BOUNDARY CONDITIONS ARTICULATED BLADE at root, Mroot=0, Yroot=0
 if bc==1,
  tip_art_bc=-1/(P(2,3)*P(4,4)-P(2,4)*P(4,3))*...
         [P(4,4) -P(2,4); -P(4,3) P(2,3)]*[Mn(nbe); Yn(nbe)];

  Ztip=[0; 0; tip_art_bc];
 end

% BOUNDARY CONDITIONS RIGID BLADE at root, THETAroot=0, Yroot=0
 if bc==2,
  tip_rig_bc=-1/(P(3,3)*P(4,4)-P(3,4)*P(4,3))*...
         [P(4,4) -P(3,4); -P(4,3) P(3,3)]*[Thetan(nbe); Yn(nbe)];

  Ztip=[0; 0; tip_rig_bc];
 end

Zroot=P*Ztip+alphaFn(:,nbe-1);

% CALCULATING Yout WHICH RECORDS THE STATES AT nbe STATIONS, FROM TIP TO ROOT

 Zn=Ztip;
```

```
  for k=1:nbe-1,
   if k==1
     Zn=Ucum(:,1:4)*Zn;
     Y=Zn+alphaFn(:,k);
     Temp=Y;
   else
     Zn=Ucum(:,4*(k-1)+1:4*(k-1)+4)*Zn;
     Temp(:,k)=Zn;
     Y(:,k)=Zn+alphaFn(:,k);
   end
  end

  Yout=[Ztip Y];
  Yout2(4*i+1:4*i+4,:)=Yout;
  Xout=[rn/12]./R;
end
output
```

APPENDIX C: OUTPUT.M

```
% Subroutine to view options
view=1;
while view > 0,
disp(' ')
disp('         *** BLADE DYNAMICS OUTPUT MENU ***')
disp(' ')
disp('       CHOOSE WHICH OUTPUT OPTION YOU WOULD LIKE')
disp(' ')
disp('    1. View the steady and first ten harmonic responses')
disp(' ')
disp('    2. View a mesh plot of the flatwise Shear, Moment, Slope and Deflection')
disp('        at all azimuth positions')
disp(' ')
disp('    3. View the flatwise Shear, Moment, Slope and Shear at a specific')
disp('        azimuth position')
disp(' ')
disp('    4. View the stiffness (EI) and weight distribution')
disp(' ')
disp('    0. Exit')
disp(' ')
disp(' *** FOR A PRINTOUT CHOOSE THE "File" OPTION IN THE DESIRED GRAPH WINDOW ***')
view=input('      >> ');
disp(' ')

% viewing the steady and the first ten harmonic responses
if view==1,
 disp(' ')
 for i=0:10,
  figure(i+1)
  subplot(2,2,1)
  plot(Xout,real(Yout2(4*i+1,:)),'b-',Xout,imag(Yout2(4*i+1,:)),'r-.');grid
  title(sprintf('SHEAR ,%3.0f HARMONIC RESPONSE',i))
  xlabel('r/R')
  ylabel('SHEAR FORCES (LBS)')

  subplot(2,2,2)
  plot(Xout,real(Yout2(4*i+2,:)),'b-',Xout,imag(Yout2(4*i+2,:)),'r-.');grid
  xlabel('r/R')
  ylabel('MOMENTS (IN-LBS)')
  title(sprintf('.....MOMENT, %3.0f HARMONIC RESPONSE',i))

  subplot(2,2,3)
  plot(Xout,(real(Yout2(4*i+3,:)))*180/pi,'b-',Xout,(imag(Yout2(4*i+3,:)))*180/pi,'r-.');grid
  title(sprintf('SLOPE,%3.0f HARMONIC RESPONSE',i))
  xlabel('r/R')
  ylabel('SLOPE (DEG)')

  subplot(2,2,4)
  plot(Xout,real(Yout2(4*i+4,:)),'b-',Xout,imag(Yout2(4*i+4,:)),'r-.');grid
  title(sprintf('DEFLECTION,%3.0f HARMONIC RESPONSE',i))
  xlabel('r/R')
  ylabel('VERT DISPLACEMENT (IN)')
```

```
  end
 end

 % viewing the mesh plot for the total response
 if view==2,
 figure(1)
 for j=1:length(psi),
   Yout3=Yout2(1:4,:); %steady state
   for i=1:10, %harmonics
     Ynew=real(Yout2(4*i+1:4*i+4,:)).*sin(psi(j)/57.3*i)+imag(Yout2(4*i+1:4*i+4,:)).*cos(psi(j)/57.3*i);
     Yout3=Yout3+Ynew;
   end
   YoutS((j-1)+1,:)=Yout3(1,:);
   YoutM((j-1)+1,:)=Yout3(2.:);
   YoutTh((j-1)+1,:)=Yout3(3,:);
   YoutY((j-1)+1,:)=Yout3(4,:);
 end

 subplot(2,2,1)
 mesh(Xout,psi,YoutS)
 title('TOTAL RESPONSE SHEAR PLOT')
 xlabel('r/R')
 ylabel('psi, deg')
 zlabel('SHEAR FORCES (LBS)')

 subplot(2,2,2)
 mesh(Xout,psi,YoutM)
 title('TOTAL RESPONSE MOMENT PLOT')
 xlabel('r/R')
 ylabel('psi, deg')
 zlabel('MOMENTS (IN-LBS)')

 subplot(2,2,3)
 mesh(Xout,psi,YoutTh)
 title('TOTAL RESPONSE SLOPE PLOT')
 xlabel('r/R')
 ylabel('psi, deg')
 zlabel('SLOPE (DEG)')

 subplot(2,2,4)
 mesh(Xout,psi,YoutY)
 title('TOTAL RESPONSE DEFLECTION PLOT')
 xlabel('r/R')
 ylabel('psi, deg')
 zlabel('VERT DISPLACEMENT (IN)')
 end

 % viewing the total response at a specific azimuth
 if view==3
   clc
   flag=1;
   pic=0;
   while flag > 0,
     pic=pic+1;
     disp(' ')
     az=input('Enter the azimuth angle at which you wish to see the total response (deg): ');
```

```
    Yout3=Yout2(1:4,:); %steady state
    for i=1:10, %harmonics
      Ynew=real(Yout2(4*i+1:4*i+4,:)).*sin(az/57.3*i)+imag(Yout2(4*i+1:4*i+4,:)).*cos(az/57.3*i);
      Yout3=Yout3+Ynew;
    end

    figure(pic)
    subplot(2,2,1)
    plot(Xout,Yout3(1,:),'b-');grid
    title(sprintf('TOTAL RESPONSE SHEAR AT%3.0f DEG',az))
    xlabel('r/R')
    ylabel('SHEAR FORCES (LBS)')

    subplot(2,2,2)
    plot(Xout,Yout3(2,:),'b-');grid
    title(sprintf('TOTAL RESPONSE MOMENT AT%3.0f DEG',az))
    xlabel('r/R')
    ylabel('MOMENTS (IN-LBS)')

    subplot(2,2,3)
    plot(Xout,Yout3(3,:)*180/pi,'b-');grid
    title(sprintf('TOTAL RESPONSE SLOPE AT%3.0f DEG',az))
    xlabel('r/R')
    ylabel('SLOPE (DEG)')

    subplot(2,2,4)
    plot(Xout,Yout3(4,:),'b-');grid
    title(sprintf('TOTAL RESPONSE DEFLECTION AT%3.0f DEG',az))
    xlabel('r/R')
    ylabel('VERT DISPLACEMENT (IN)')

    disp(' ')
    disp('Do you want to see another azimuth angle?')
    flag=input('  0) No       1) Yes  >>');
  end
end
  if view==4
    clc
    subplot (2,1,1)
    plot(rn,EI./1e6);grid
    title(['STIFFNESS DISTRIBUTION FOR ',filename3])
    xlabel('BLADE STATION, IN')
    ylabel('FLATWISE STIFFNESS, EIxx, LB*IN^2x10^6')

    subplot (2,1,2)
    plot(rn,Wn);grid
    title(['WEIGHT DISTRIBUTION FOR ',filename3])
    xlabel('BLADE STATION, IN')
    ylabel('WEIGHT, LB/IN')
  end
end
```

APPENDIX D: H2 SAMPLE HELICOPTER

The following is a listing of the performance data file which contains both input and calculated parameters based on the sample helicopter "H2", taken from Ref. 5.

filename: h2_3.prf

Forward velocity = 150 kts
Temperature = 59 degs F
Pressure altitude = 0 ft
Gross weight = 33000 lbs
Number of blades = 6
Rotor radius = 36.00 ft
Blade mean chord = 1.97 ft
Blade twist = -6.00 degs
Blade lift curve slope = 5.73
Blade weight = 246.43 lbs
Rotational velocity = 19.37 rads/sec
Blade grip length = 9.92 ft
Hinge offset = 2.00 ft
Equivalent flat plate area = 44.00 ft^2
Vertical projected area = 495.00 ft^2
Wing area = 0.00 ft^2
Wing span = 0.00 ft
Wing CL = 0.00
Wing CDo = 0.0000
Wing efficiency factor = 0.00
Horizontal tail area = 75.00 ft^2
Horizontal tail span = 15.00 ft
Horizontal tail CL = 0.40
Horizontal tail CDo = 0.0115
Vertical tail area = 84.00 ft^2
Vertical tail span = 14.00 ft
Vertical tail CL = 0.40
Vertical tail CDo = 0.0114
Fuselage drag = 3356 lbs
Rotor drag = 482 lbs
Wing lift = 0 lbs
Wing drag = 0 lbs
Horizontal tail lift = 2288 lbs
Horizontal tail drag = 187 lbs
Vertical tail side force = 2563 lbs
Vertical tail drag = 248 lbs
Auxiliary thrust = 0 lbs
Tip path angle = 7.90 degs
Rotor coning angle = 6.65 degs
Location of mean thrust (r/R) = 0.73
Collective pitch at .7 r/R = 9.23 degs
1st lat cyclic term-A1 (deg) = 2.87
1st long cyclic term-B1 (deg) = -5.75

solidity = 0.105
CT/sigma = 0.063
CQ/sigma = 0.0052
CH/sigma = 0.0010
Tip mach of the adv. blade = 0.845
Advance ratio = 0.360
Rotor thrust required (TPP) = 31006 lbs
Rotor power required = 3237 h.p.
Rotor torque = 91908 ft-lbs

The following is a listing of the blade data file which contains the material properties of the "H2" articulated rotor system, taken from Ref. 5 pg. 203

RADII (IN.)	CHORD (IN.)	SEGMENT WEIGHT (LB.)	FLATWISE Ixx ($IN.^4$)
432.00	23.65	6.97	1.87
410.40	23.65	19.60	3.75
388.80	23.65	14.51	4.20
367.20	23.65	15.77	4.75
345.60	23.65	15.74	5.05
324.00	23.65	17.14	5.42
302.40	23.65	10.29	5.58
280.80	23.65	17.63	5.68
259.20	23.65	17.02	5.80
237.60	23.65	18.04	5.95
216.00	23.65	17.71	6.05
194.40	23.65	16.61	6.18
172.80	23.65	16.42	6.25
151.20	23.65	16.42	6.40
129.60	23.65	17.33	6.45
108.00	9.71	18.46	8.10
86.40	9.71	38.11	24.10
64.80	9.71	124.62	100.00
43.20	9.71	77.12	100.00
24.00	9.71	84.59	100.00

LIST OF REFERENCES

1. Nicholson, R. K., *Computer Code for Interactive Rotorcraft Preliminary Design Using a Harmonic Balance Method for Rotor Trim,* Naval Postgraduate School, September 1993.

2. Wirth, W. M., *Linear Modeling of Rotorcraft for Stability and Control Analysis and Preliminary Design,* Naval Postgraduate School, September, 1993.

3. Gerstenberger, W. and Wood, E. R., *Analysis of Helicopter Aeroelastic Characteristics in High-Speed Flight,* AIAA Journal, Vol. 1, no. 10, October 1963.

4. Prohl, M. A., *A General Method for Calculating Critical Speeds of Flexible Rotors,* Journal of Applied Mechanics, Vol. 12, no. 3, September 1945.

5. Myklestad, N. O., *A New Method of Calculating Natural Modes of Uncoupled Bending Vibrations of Airplane Wings and Other Types of Beams,* Journal of the Aeronautical Sciences, Vol. 11, no. 4, April 1944.

6. Holzer, H., *Die Berechnung der Drehschwinggungen,* Julius Springer, Berlin, 1922.

7. Hurty, W. C. and Rubinstein, M. F., *Dynamics of Structures*, Prentice-Hall, Inc., 1964.

8. Young, D., *Vibrations of Multi-Degree of Freedom Systems and Continuous Systems,* General Electric Company, General Engineering Laboratory, 1955.

9. Wood, E. R., and Buffalano, A. C., *Parametric Investigation of the Aerodynamic and Aeroelastic Characteristics of Articulated and Rigid (Hingeless) Helicopter Rotor Systems,* TRECOM Technical Report 64-15, April 1964.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52 2
   Naval Postgraduate School
   Monterey, California 93943-5002

3. Chairman, Code AA 1
   Department of Aeronautics and Astronautics
   Naval Postgraduate School
   Monterey, California 93943-5000

4. Professor E. Roberts Wood 4
   Department of Aeronautics and Astronautics
   Code AA/Wd
   Naval Postgraduate School
   Monterey, California 93943-5000

5. Professor Sheshagiri K. Hebbar 1
   Department of Aeronautics and Astronautics
   Code AA/Hb
   Naval Postgraduate School
   Monterey, California 93943-5000

6. Lt. Juan D. Cuesta, USNR 2
   45 Sherbrooke Drive
   Smithtown, New York 11787

7. Maj. Robert K. Nicholson, Jr.,USA 1
   1116 Walker Road
   Great Falls, Virginia 22066

8. Maj. Walter M. Wirth, Jr., USA 1
   6112 Pueblo Court
   Burke, Virginia 22066

9. Lt. Mark Couch, USN 1
928 Grandy Drive
Chesapeake, Virginia 23322

10. Lt. Dale Feddersen, USN 1
117 North Harmony Street
West Salem, Wisconsin 54669